А. И. МАЛЬЦЕВ

АЛГОРИТМЫ И РЕКУРСИВНЫЕ ФУНКЦИИ



Данная книга посвящена систематическому изложению теории алгоритмов и рекурсивных функций. В начальных главах указываются точные определения различных классов рекурсивных функций, доказываются фундаментальные свойства этих классов и строятся примеры рекурсивных функций, обладающих рядом особо важных свойств. В последующих главах рассматриваются классы алгоритмов, достаточные для вычисления произвольных рекурсивных функций и связанные с так называемыми машинами Тьюринга—Поста. Несколько параграфов посвящены приложениям теории алгоритмов к алгебре (проблема эквивалентности слов в конечно определенных полугруппах), математической логике (тождественно истинные формулы языка 1-й ступени) и теории чисел (проблема разрешимости диофантовых уравнений).

Книга возникла из курса лекций, читанных автором в Новосибирском университете. Она может служить пособием при первоначальном изучении предмета. В книге содержится много дополнительного материала, появившегося в журналах лишь в последние годы. Это может сделать ее интересной и для аспирантов и научных работников, занимающихся как математической логикой, так и ее приложениями в математике, теории программирования, математической лингвистике и в других смежных областях науки.

А. И. МАЛЬЦЕВ

АЛГОРИТМЫ И РЕКУРСИВНЫЕ ФУНКЦИИ



А. И. МАЛЬЦЕВ

АЛГОРИТМЫ И РЕКУРСИВНЫЕ ФУНКЦИИ



ИЗДАТЕЛЬСТВО «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
МОСКВА 1965

518
М 21
УДК 519.95

$\frac{2-2-3}{43-65}$

ОГЛАВЛЕНИЕ

Глава VI

Варианты машин и алгоритмов Тьюринга — Поста

ПРЕДИСЛОВИЕ

Еще в 30-х годах нашего столетия математическая логика и возникавшая тогда теория алгоритмов казались наиболее абстрактными и наиболее далекими от практических приложений математическими дисциплинами. В настоящее время положение коренным образом изменилось. Ныне общепризнанно, что обе названные дисциплины образуют теоретический фундамент для создания и применений быстродействующих вычислительных и управляющих систем. Резко возрос удельный вес математической логики и теории алгоритмов и в самой математике. Более того, в значительной степени через теорию алгоритмов и математическую логику происходит ныне проникновение математических методов в биологию, лингвистику, экономику вплоть до философии естествознания. Все это привело к тому, что математическая логика и теория алгоритмов начали проникать и в учебные планы наших университетов и désinститутов в качестве дисциплин, обязательных для изучения студентами-математиками всех специальностей.

Настоящая книга возникла в результате обработки конспектов лекций по математической логике, теории алгоритмов и их приложений, читавшихся автором в 1956—1959 гг. в Ивановском педагогическом институте и с 1960 г. в Новосибирском университете. В ней излагается лишь общая теория алгоритмов и рекурсивных функций.

Целиком за пределами книги остались массивы теории автоматов, приложений теории алгоритмов к формальным теориям, теории степеней неразрешимости. Сколько-нибудь подробное изложение этих дисциплин в настоящее время требует специальных монографий.

Более серьезным недостатком предлагаемой книги может оказаться отсутствие в ней сведений о реальных вычислительных машинах. Однако в настоящее время теория программирования и принципы устройства реальных вычислительных машин читаются во всех университетах в качестве самостоятельных курсов. По этим курсам написано много учебников самого различного уровня. Поэтому нам казалось излишним включать соответствующие вопросы в общую теорию алгоритмов и в данной книге они даже не упоминаются.

Так как лекции по теории алгоритмов иногда читаются до лекций по математической логике, то логическая символика используется нами в очень ограниченном размере и значения всех употребляемых логических символов подробно разъясняются. По этой же причине в книге не обсуждаются вопросы, связанные с интуиционистским и конструктивистским истолкованиями результатов.

Как обычно, от читателей никаких предварительных специальных знаний, выходящих за пределы программы средней школы, не требуется. Доказательства всюду проведены полностью за исключением последних глав, где иногда опущены рассуждения рутинного характера, которые легко восстановит каждый читатель, добравшийся до этих глав.

Первая половина книги в несколько ином виде была издана студентами Новосибирского университета в 1960 г. ротапринтным способом. Остальные главы читались в рукописи сотрудниками и студентами кафедры алгебры и математической логики НГУ. Всем им автор признателен за советы и замечания.

А. И. Мальцев

ВВЕДЕНИЕ

Уже на самых ранних ступенях развития математики (Древний Египет, Вавилон, Греция) в ней стали возникать различные вычислительные процессы чисто механического характера; с их помощью искомые величины ряда задач вычислялись последовательно из данных исходных величин по определенным правилам и инструкциям. Со временем все такие процессы в математике получили название алгоритмов или — по другим написаниям — алгорифмов. Вплоть до 30-х годов нашего столетия это понятие алгоритма в основе своей не менялось, хотя приобретало все бóльшую и бóльшую отчетливость. Тем не менее, это понятие оставалось интуитивным понятием, имевшим скорее методологическое, а не математическое значение. Сущность его легко уясняется из следующих примеров.

В десятичной системе счисления натуральные числа изображаются конечными последовательностями цифр 0, 1,. . ., 9. Спрашивается, как найти десятичную запись числа, равного сумме, разности, произведению, частному двух чисел, заданных своими десятичными записями? Известные всем из начальной школы процессы, с помощью которых решаются эти задачи, и являются алгоритмами сложения, вычитания, умножения и деления целых чисел в десятичной системе счисления. Аналогичные алгоритмы известны и для произвольных p-ичных систем счисления.

Таким же ярким примером алгоритма может служить и процесс нахождения наибольшего общего делителя двух положительных натуральных чисел a_1, a_2. Состоит он, как известно, в следующем:

1) делим a_1 на a_2, находим остаток a_3 и смотрим, равен он 0 или нет. Если равен, то процесс обрывается и a_2 — искомый наибольший делитель. Если же $a_3 > 0$, то

2) делим a_2 на a_3 и находим остаток a_4. Если $a_4 = 0$, то процесс обрывается и a_3 — искомый наибольший общий делитель. Если $a_4 > 0$, то

3) делим a_3 на a_4 и т. д.

Так как $a_2 > a_3 > a_4 > \ldots \geqslant 0$, то указанный процесс оборвется самое большее через a_2 шагов и мы при его обрыве найдем требуемый наибольший общий делитель чисел a_1 и a_2.

Аналогичные процессы применяются для нахождения наибольшей общей меры двух отрезков, наибольшего общего делителя двух многочленов и т. п. Все эти процессы известны сейчас под именем *алгоритмов Евклида*.

Из других алгоритмов укажем еще алгоритмы разложения натурального числа на простые множители, извлечения квадратного корня из натурального числа, процесс решения системы линейных уравнений методом последовательного исключения неизвестных и т. д.

Отметим теперь несколько общих черт алгоритмов, ясно вырисовывающихся из предыдущих примеров и часто признающихся (Колмогоров и Успенский [45], Марков [55]) характерными для понятия алгоритма.

а) Алгоритм — это процесс последовательного построения величин, идущий в дискретном времени таким образом, что в начальный момент задается исходная конечная система величин, а в каждый следующий момент система величин получается по определенному закону (программе) из системы величин, имевшихся в предыдущий момент времени (*дискретность алгоритма*).

б) Система величин, получаемых в какой-то (не начальный) момент времени, однозначно определяется системой величин, полученных в предшествующие моменты времени (*детерминированность алгоритма*).

в) Закон получения последующей системы величин из предшествующей должен быть простым и локальным (*элементарность шагов алгоритма*).

г) Если способ получения последующей величины из какой-нибудь заданной величины не дает результата, то должно быть указано, что надо считать результатом алгоритма (*направленность алгоритма*).

д) Начальная система величин может выбираться из некоторого потенциально бесконечного множества (*массовость алгоритма*).

Понятие алгоритма, в какой-то мере определяемое требованиями а) — д), конечно, не строгое: в формулировках этих требований встречаются слова «способ», «величина», «простой», «локальный», точный смысл которых не установлен. В дальнейшем это нестрогое понятие алгоритма будет называться непосредственным или интуитивным понятием алгоритма.

Интуитивное понятие алгоритма хотя и нестрогое, но настолько ясное, что практически не было серьезных случаев, когда математики разошлись бы в мнениях относительно того, является ли алгоритмом тот или иной конкретно заданный процесс. В этом обстоятельстве легко найти объяснение своеобразного положения, сложившегося в математике с алгоритмическими проблемами к началу XX в. В этих проблемах требуется найти алгоритм для решения некоторой совокупности родственных задач, в условия которых входит конечная система параметров, могущих принимать обычно произвольные целочисленные значения. Например, требуется найти алгоритм, позволяющий для каждой четверки целых чисел a, b, c, d узнать, существуют или нет целые числа x, y, удовлетворяющие уравнению

$$ax^2 + bxy + cy^2 = d.$$

Был найден процесс, при помощи которого, исходя из заданных чисел, через конечное число «шагов» операций можно было на указанный вопрос получить ответ «да» или «нет». Многие другие алгоритмические проблемы также были решены путем указания конкретных разрешающих процессов. Положение существенно изменилось в XX в., выдвинувшем на первый план такие алгоритмические проблемы, существование положительного решения которых было сомнительным. Действительно, одно дело — доказать существование алгоритма, другое — доказать отсутствие алгоритма. Первое можно сделать путем фактического описания процесса, решающего задачу; в этом случае достаточно и интуитивного понятия алгоритма, чтобы удостовериться в том, что описанный процесс есть алгоритм.

Доказать несуществование алгоритма таким путем невозможно. Для этого надо точно знать, что такое алгоритм. В двадцатых годах нашего века задача точного определения понятия алгоритма стала одной из центральных математических проблем. Решение ее было получено в середине тридцатых годов в работах Гильберта, Геделя, Чёрча, Клини, Поста и Тьюринга в двух формах. Первое решение было основано на понятии рекурсивной функции, второе — на описании точно очерченного класса процессов.

Выше уже говорилось, что для алгоритмических проблем типичным является положение, когда требуется найти алгоритм для решения задачи, в условия которой входят значения некоторой конечной системы целочисленных параметров $x_1, \ldots, x_n$, а искомым результатом служит также целое число y. Иначе говоря, речь идет о существовании алгоритма для вычисления значений числовой функции y, зависящей от целочисленных значений аргументов $x_1, \ldots$ $\ldots, x_n$. Числовые функции, значения которых можно вычислять посредством некоторого (единого для данной функции) алгоритма, называются *вычислимыми функциями*. Поскольку понятие алгоритма в этом определении берется в интуитивном смысле, то и понятие вычислимой функции оказывается интуитивным. Тем не менее, при переходе от алгоритмов к вычислимым функциям возникает одно очень существенное обстоятельство. Совокупность процессов, удовлетворяющих условиям а) — д) и, следовательно, подпадающих под интуитивное понятие алгоритма, очень обширна и мало обозрима. Напротив, совокупность вычислимых функций для самых разных пониманий процессов, удовлетворяющих условиям а) — д), оказалась одной и той же и притом легко описываемой в обычных математических терминах. Эта точно описанная совокупность числовых функций, совпадающая с совокупностью всех вычислимых функций при самом широком до сих пор известном понимании алгоритма, носит название совокупности *рекурсивных функций*.

Отправляясь от идей Гильберта, Гедель [14] впервые описал класс всех рекурсивных функций как класс всех числовых функций, определимых в некоторой формальной системе. Исходя из совершенно других предпосылок, Чёрч в 1936 г. пришел к тому же классу числовых

функций, что и Гедель. Анализ идей, приведших к этому классу функций, позволил Чёрчу первым опубликовать гипотезу о том, что класс рекурсивных функций тождествен с классом всюду определенных вычислимых функций. Эта гипотеза ныне известна под именем *тезиса Чёрча.* Так как понятие вычислимой функции точно не определяется, то тезис Чёрча доказать нельзя.

Выше уже отмечалось, что, перерабатывая некоторые исходные данные $x_1, \ldots, x_n$ согласно заданному алгоритму, мы можем встретиться с явлением, что процесс переработки никогда не оканчивается. В этом случае говорят, что данный алгоритм перерабатывает числа $x_1, \ldots, x_n$ в «неопределенность». Для того чтобы охватить и эти весьма важные случаи бесконечно длящихся процессов переработки, Клини [41] ввел понятие частично рекурсивной функции и высказал гипотезу, что все частичные (т. е. не обязательно определенные для всех значений аргументов) функции, вычислимые посредством алгоритмов, являются частично рекурсивными. Конечно, более общая гипотеза Клини так же недоказуема, как и гипотеза Чёрча. Однако исследования, проводившиеся весьма многими математиками в течение последних тридцати лет, выявили полную целесообразность считать понятие частично рекурсивной функции научным эквивалентом интуитивного понятия вычислимой частичной функции.

В дальнейшем мы не будем различать тезисы Чёрча и Клини и под именем тезиса Чёрча будем понимать гипотезу Чёрча в том расширенном виде, который был придан ей Клини.

Тезис Чёрча оказался достаточным, чтобы придать необходимую точность формулировкам алгоритмических проблем и в ряде случаев сделать возможным доказательство их неразрешимости. Причина этого заключается в том, что обычно в алгоритмических проблемах математики речь идет не об алгоритмах, а о вычислимости некоторых специальным образом построенных функций. В силу тезиса Чёрча вопрос о вычислимости функции равносилен вопросу о ее рекурсивности. Понятие рекурсивной функции строгое. Поэтому обычная математическая техника позволяет иногда непосредственно доказать, что решающая задачу функция не может быть рекурсивной. Именно этим путем

Чёрчу [115] удалось первому доказать неразрешимость основной алгоритмической проблемы логики предикатов — проблемы тождественной истинности формул исчисления первой ступени.

Точное описание класса частично рекурсивных функций вместе с тезисом Чёрча дает одно из возможных решений задачи об уточнении понятия алгоритма. Однако это решение не вполне прямое, так как понятие вычислимой функции является вторичным по отношению к понятию алгоритма. Спрашивается, нельзя ли уточнить непосредственно само понятие алгоритма и уже затем при его помощи определить точно и класс вычислимых функций? Это было сделано Постом [70] и Тьюрингом [101] независимо друг от друга и почти одновременно с изложенными выше работами Чёрча и Клини. Основная мысль Поста и Тьюринга заключалась в том, что алгоритмические процессы — это процессы, которые может совершать подходяще устроенная «машина». В соответствии с этой мыслью ими были описаны в точных математических терминах довольно узкие классы машин, но на этих машинах оказалось возможным осуществить или имитировать все алгоритмические процессы, которые фактически когда-либо описывались математиками. Алгоритмы, осуществимые на упомянутых машинах, было предложено рассматривать как математических «представителей» вообще всех алгоритмов. Несложные выкладки показали, что класс функций, вычислимых на этих машинах, в точности совпадает с классом всех частично рекурсивных функций. Тем самым было получено еще одно фундаментальное подтверждение тезиса Чёрча.

Машины, введенные Постом и Тьюрингом, отличались не очень существенно и в дальнейшем стали называться машинами Тьюринга, несмотря на то, что были введены указанными авторами одновременно и независимо друг от друга. Более того, машины, описанные в данной книге в главе V под именем машин Тьюринга, почти точно совпадают с тем вариантом, который был предложен Постом.

Сравнивая обычное определение частично рекурсивных функций с определением тех же функций как функций, вычислимых на машинах Тьюринга — Поста, легко заме-

тить следующую направленность этих определений. Обычное определение частично рекурсивных функций настолько широко, что из него почти непосредственно видна частичная рекурсивность функций, вычислимых посредством процессов, алгоритмический характер которых интуитивно ясен. Напротив, определение с помощью машин Тьюринга — Поста очень специальное. Его цель — показать, как самые сложные процессы можно моделировать на весьма простых устройствах. Поэтому, если надо показать, что алгоритмические процессы можно моделировать на еще каких-либо специальных устройствах, то машины Тьюринга — Поста обычно берутся в качестве отправного пункта рассуждений. В частности, этим путем была решена П. С. Новиковым [66] классическая проблема тождества теории групп. Это была первая крупная алгоритмическая проблема, возникшая в математике независимо от математической логики и теории алгоритмов и решенная при помощи развитой теории алгоритмов. С другой стороны, одна из наиболее знаменитых алгоритмических проблем математики — так называемая 10-я проблема Гильберта о разрешимости алгебраических уравнений в целых числах — пока все еще остается открытой (1963 г.), хотя алгоритмическая неразрешимость близкой задачи о существовании целочисленных решений показательных уравнений была недавно установлена группой американских математиков (см. § 16 этой книги).

Первоначально теория алгоритмов возникла в связи с внутренними потребностями теоретической математики. Математическая логика, основания математики, алгебра, геометрия и анализ остаются и сегодня одной из основных областей приложения теории алгоритмов. Другая область ее приложений возникла в сороковых годах в связи с созданием быстродействующих электронных вычислительных и управляющих машин, в которых с большой точностью моделируются машины Тьюринга — Поста. Наконец, теория алгоритмов оказалась тесно связанной и с рядом областей лингвистики, экономики, физиологии мозга и психологии, философии естествознания. Например, достаточно старые вопросы о том, является ли мозг сложной машиной, как он работает, в чем состоит различие между трудом творческим и механическим, в первом

приближении теперь часто формулируют следующим образом: можно ли принципиально построить машину Тьюринга, перерабатывающую поступающую информацию так же, как мозг какого-нибудь животного, можно ли отождествлять труд механический с трудом согласно заданному алгоритму и т. п. Ясно, что ответ на эти и многие другие аналогичные вопросы сама теория алгоритмов дать не в состоянии, но она помогает более отчетливо понять суть вопросов такого рода.

Чтобы иметь возможность более уверенно решать алгоритмические задачи, возникающие в различных отделах теоретической и прикладной математики, необходимо иметь достаточно развитую самостоятельную теорию алгоритмов. В настоящее время такая независимая и богатая теория алгоритмов уже создана. Первое систематическое изложение ее было осуществлено С. Клини [43], монография которого «Введение в метаматематику» остается и поныне основным руководством в рассматриваемой области науки. Однако теория алгоритмов и рекурсивных функций изложена в этой монографии в тесном переплетении с рядом разделов математической логики с расчетом на очень квалифицированного читателя. Кроме того, со времени выхода в свет указанной монографии появилось много новых важных работ по теории алгоритмов и рекурсивных функций, рассеянных по различным журналам и часто отсутствующих в библиотеках.

В настоящей книге, в первую очередь, изложены все основные результаты, относящиеся собственно к теории алгоритмов и рекурсивных функций. Из приложений рассмотрено лишь несколько частных задач, решение которых почти непосредственно вытекает из теорем и методов, изложенных в основном тексте.

О логической зависимости разделов книги

В данной книге обычным шрифтом напечатаны те разделы, материал которых излагается, как правило, в общих курсах теории алгоритмов. Мелким шрифтом набраны разделы, содержащие результаты более специального характера. При первом чтении книги эти разделы, при желании,

можно опустить. Наконец, много важных результатов специального характера читатель может найти в дополнениях, примерах и задачах, помещенных в конце каждого параграфа. Результаты эти приведены без доказательств, но указаны источники, где доказательства можно найти.

При чтении этой книги не обязательно изучать ее разделы в той последовательности, в какой они напечатаны. Более того, лицам, интересующимся в первую очередь лишь приложениями теории алгоритмов, будет, вероятно, удобнее изучать разделы книги в следующем порядке: § 1, § 2, пп. 3.1—3.4, § 4, § 11, пп. 12.1—12.4, п. 6.3, п. 12.5, § 13, пп. 7.1—7.3, пп. 8.1—8.2, § 14, § 15, § 16.

ГЛАВА I

ОСНОВНЫЕ ПОНЯТИЯ

Во введении подробно обсуждалось значение ряда основных понятий: частичной функции, алгоритма, алгоритмически вычислимой функции и других. Однако перечисленные понятия были введены лишь описательно, без отчетливых и точных определений. Цель данной главы — указать точные определения для некоторых из этих понятий и прежде всего для понятия рекурсивной функции и тем самым заложить прочный фундамент для более детального изучения этих понятий в последующих главах книги.

§ 1. ФУНКЦИИ И ОПЕРАЦИИ

Этот параграф имеет вспомогательное значение. Здесь с целью установления терминологии и употребляющейся далее символики напоминаются определения ряда всем известных общематематических понятий.

1. 1. Алфавит. Слова. Исходным материалом для нас будет служить понятие ленты, разделенной на («равные») участки, называемые ячейками или клетками. Лента будет считаться конечной длины в каждый момент времени, неограниченно продолжаемой (надстраиваемой) в обе стороны и направленной, так что у каждой ячейки есть соседняя справа и соседняя слева.

Предполагается, что каждая ячейка ленты может находиться в различных состояниях и что эти состояния сравнимы, так что мы можем без какой-либо двойственности решить относительно состояний произвольных двух ячеек, находятся ли они в «одинаковых» состояниях или в разных. Одно из возможных состояний ячеек будет называться исходным. Ячейки, находящиеся в этом состоя-

нии, называются пустыми. Остальные состояния будут обозначаться буквами, занимающими соответствующие ячейки.

Произвольная конечная совокупность букв называется *алфавитом*. Выражение: «рассмотрим алфавит, состоящий из букв A, B» означает, что будет рассматриваться лента, ячейки которой могут находиться в состояниях, условно обозначаемых символами A, B.

Последовательность ячеек, занятых некоторыми буквами, называется *словом*. Словом в данном алфавите называется каждое слово, все буквы которого принадлежат этому алфавиту. Например, если алфавит состоит из букв A, B, C, то слова

$$A,\ BAA,\ CA,\ CC,\ ACBAB \tag{1}$$

будут словами в этом алфавите. В то же время первые два из них будут словами и в алфавите A, B.

Длиной слова называется число входящих в него символов (т. е. число занятых им ячеек). Так, длинами написанных выше слов (1) являются соответственно числа 1, 3, 2, 2, 5. Два слова называются (графически) равными, если они имеют одинаковые длины и на соответствующих местах их находятся равные буквы.

Пусть символы $\mathfrak{a}$, $\mathfrak{b}$ обозначают слова, записанные в алфавите, не содержащем самих символов $\mathfrak{a}$, $\mathfrak{b}$. Тогда через $\mathfrak{a}\mathfrak{b}$ обозначается слово, получающееся, если сначала выписать слово $\mathfrak{a}$, а затем приписать справа к нему слово $\mathfrak{b}$. Например, если $\mathfrak{a}$ означает слово aba, а $\mathfrak{b}$ – слово ac, то $\mathfrak{a}\mathfrak{b}$ будет обозначать слово $abaac$. Слово $\mathfrak{a}\mathfrak{b}$ называется *композицией* (иногда произведением) слов $\mathfrak{a}$ и $\mathfrak{b}$. Операция композиции слов, очевидно, ассоциативна, но не коммутативна.

По чисто формальным причинам удобно ввести еще понятие пустого слова (не следует смешивать с пустой ячейкой), композиция которого с любым словом считается по определению равной этому последнему слову.

Слово $\mathfrak{a}$ называется *подсловом* слова $\mathfrak{b}$, если $\mathfrak{b}$ можно представить в виде

$$\mathfrak{b} = \mathfrak{c}\mathfrak{a}\mathfrak{d}, \tag{2}$$

где $\mathfrak{c}$, $\mathfrak{d}$ — подходящие (возможно пустые) слова. Если $\mathfrak{a}$ — подслово слова $\mathfrak{b}$, то говорят также, что $\mathfrak{a}$ входит в $\mathfrak{b}$.

При заданных словах $\mathfrak{b}$, $\mathfrak{a}$ разложений вида (2) может быть несколько. Например, для слов

$$\mathfrak{b} = ababab, \quad \mathfrak{a} = ab$$

существуют следующие разложения указанного вида:

$$\mathfrak{b} = \mathfrak{a} \cdot abab = ab \cdot \mathfrak{a} \cdot ab = abab \cdot \mathfrak{a}.$$

Если в разложении вида (2) слово $\mathfrak{c}$ имеет наименьшую возможную длину, то говорят о первом *вхождении* слова $\mathfrak{a}$ в $\mathfrak{b}$. Аналогичным образом можно говорить о втором вхождении слова $\mathfrak{a}$ в слово $\mathfrak{b}$ и т. д. Если слово $\mathfrak{a}$ есть просто буква, то говорят кратко о вхождениях буквы $\mathfrak{a}$ в слово $\mathfrak{b}$.

Пусть разложение (2) соответствует первому (вообще k-му) вхождению слова $\mathfrak{a}$ в слово $\mathfrak{b}$ и пусть $\mathfrak{m}$ — какое-нибудь слово. Тогда слово $\mathfrak{c}\mathfrak{m}\mathfrak{d}$ называют словом, полученным заменой в слове $\mathfrak{b}$ первого (k-го) вхождения слова $\mathfrak{a}$ словом $\mathfrak{m}$. В этом случае говорят также о подстановке в слово $\mathfrak{b}$ слова $\mathfrak{m}$ вместо соответствующего вхождения слова $\mathfrak{a}$. Например, производя подстановку слова ac в слово $ababab$ вместо второго вхождения в него слова ab, получим слово $abacab$. Подставляя в то же слово вместо первого вхождения слова ab пустое слово, получим слово $abab$.

1.2. Функции. Термы. Пусть A, B — какие-либо множества, например, совокупности всех слов в подходящих алфавитах. Совокупность всех пар вида $\langle a, b \rangle$, где $a \in A$, $b \in B$, называется декартовым произведением A на B и обозначается через $A \times B$. Аналогично, если A_1, A_2, ..., A_m — некоторые множества, взятые в определенной последовательности, то совокупность $(((A_1 \times A_2) \times A_3) \times \ldots \times A_m)$ называется декартовым произведением указанных множеств и обозначается кратко через $A_1 \times A_2 \times \ldots \times A_m$. Если все множества A_1, ..., A_m совпадают друг с другом, то их декартово произведение называется декартовой m-й степенью множества A.

Если некоторым элементам множества X поставлены в соответствие однозначно определенные элементы множества Y, то говорят, что задана *частичная функция* из X в Y. Совокупность тех элементов множества X, у которых есть соответствующие в Y, называется *областью определенности* функции, а совокупность тех элементов

множества Y, которые соответствуют некоторым элементам множества X, называется совокупностью значений функции. Если область определенности функции из X в Y совпадает с множеством X, то функция называется всюду определенной.

Если функция f ставит в соответствие элементу $x \in X$ элемент $y \in Y$, то y называется значением функции f в точке x. Функции f и g из X в Y называются равными, если они имеют одну и ту же область определенности и значения их совпадают в каждой точке области определенности.

Наряду с частичными функциями, имеющими непустую область определенности, во многих случаях бывает необходимо рассматривать и функцию, нигде не определенную.

Частичные функции из $X \times X \times \ldots \times X = X^{(n)}$ в Y называются частичными функциями от n переменных или n-местными функциями из X в Y. Частичная функция из $X^{(n)}$ в X называется n-местной частичной операцией на X.

Для записи функций и изучения их свойств пользуются особым формальным языком. Алфавит этого языка состоит из символов, разбитых на три группы. Символы первой группы называются предметными символами. В качестве таких символов обычно употребляются буквы $a, b, x, y, \ldots$ или те же буквы с индексами: $a_0, a_1, x_1, y_0, y_1, \ldots$

Символы второй группы называются функциональными. Это будут буквы с верхними и, возможно, нижними индексами: $f^1, g^2, f_0^1, f_2^2, \ldots$ Буква с верхним индексом n $(n \geqslant 1)$ будет называться n-местным функциональным символом. Иногда верхние индексы не пишут, но непременно указывают число мест для данных функциональных символов.

Наконец, символы третьей группы — это символы левой, правой скобок и запятой: (,) ,'.

Слова особого вида, записанные в этом функциональном алфавите, называются *термами*. Определение их следующее. Термами длины 1 называются однобуквенные слова из букв, являющихся предметными переменными. Далее пользуемся индукцией. Пусть для некоторого $s > 1$ термы длины, меньшей s, определены. Тогда слово длины s называется термом, если оно имеет вид $f(\mathfrak{a}_1, \ldots, \mathfrak{a}_n)$, где $\mathfrak{a}_1, \ldots, \mathfrak{a}_n$ — некоторые термы меньшей длины, а f — n-членный функциональный символ. Например, если a, x — предметные символы, а f, g — соответственно одноместный

и двуместный функциональные символы, то слова

$$x,\ f(a),\ g(x,\ a),\ g(f(x),\ g(a,\ x))$$

являются термами длины соответственно 1, 4, 6, 14.

Введем теперь понятие значения терма при заданных значениях предметных и функциональных символов. По определению, задать значение предметного символа — это значит указать некоторое непустое множество, называемое основным множеством, и сопоставить с рассматриваемым символом некоторый элемент этого множества. Этот элемент и называется значением данного символа.

Задать значение n-местного функционального символа — это значит сопоставить с ним какую-нибудь частичную n-местную операцию, определенную на основном множестве.

Значением терма длины 1 называется значение составляющего этот терм предметного символа. Пусть рассматривается терм $\mathfrak{a}$ большей длины. Предположим, что основное множество X выбрано, значения всех входящих в терм предметных и функциональных символов на этом множестве определены, и терм $\mathfrak{a}$ имеет вид $f(\mathfrak{a}_1, \ldots, \mathfrak{a}_n)$, где f — n-членный функциональный символ, а $\mathfrak{a}_1, \ldots, \mathfrak{a}_n$ — термы меньшей длины. По индукции мы можем считать, что значения термов $\mathfrak{a}_1, \ldots, \mathfrak{a}_n$ уже определены и равны некоторым элементам $x_1, \ldots, x_n$ множества X. По условию символу f поставлена в соответствие n-местная операция f_0, определенная на X. Значение этой операции в точке $\langle x_1, \ldots, x_n \rangle$ будет каким-то элементом x из X. Этот элемент x и будет значением терма $\mathfrak{a}$ при заданных значениях предметных и функциональных символов.

Может случиться, что в качестве значений функциональных символов взяты частичные операции, определенные не всюду на основном множестве. Тогда и значение терма может оказаться неопределенным. А именно, значение терма $\mathfrak{a} = f(\mathfrak{a}_1, \ldots, \mathfrak{a}_n)$ считается неопределенным, если значение хотя бы одного из термов $\mathfrak{a}_1, \ldots, \mathfrak{a}_n$ не определено или не определено значение операции f_0 в точке $\langle x_1, \ldots, x_n \rangle$, где $x_1, \ldots, x_n$ — значения термов $\mathfrak{a}_1, \ldots, \mathfrak{a}_n$. Например, термы

$$(2+8):2, \quad (2\times 7)-7$$

имеют соответственно значения 5, 7, а значения термов

$$5-(3+3),\ 2:3,\ 3:(2-2),\ 5+(2:3)$$

не определены. Терм

$$((3x+y):y)+4$$

имеет значение 7 при значениях 2 и 3 для предметных символов x, y, не определен при значениях 2, 5 и имеет значение 5 при значениях 0 и 1 символов x, y.

Имея несколько операций, заданных на каком-то множестве X, можно при помощи термов записать бесконечное множество новых операций, определенных на том же множестве X. Делается это следующим образом. Обозначаем заданные операции какими-либо функциональными символами $f_1, \ldots, f_s$. Пусть, кроме того, в X выделены еще некоторые элементы, обозначаемые символами $a_1, \ldots, a_r$. Рассмотрим произвольный терм $\mathfrak{a}$, составленный из символов $a_1, \ldots, a_r$, $f_1, \ldots, f_s$ и вспомогательных предметных символов $x_1, \ldots, x_t$. Так как основное множество X и значения символов $a_1, \ldots, a_r$, $f_1, \ldots, f_s$ уже фиксированы, то, задавая по произволу значения для символов $x_1, \ldots, x_t$ в множестве X, мы можем найти значение терма $\mathfrak{a}$ (которое может оказаться и неопределенным, если некоторые из заданных операций определены не всюду). Таким образом, каждой последовательности элементов $\langle x_1, \ldots, x_t\rangle$ множества X отвечает определенное (или неопределенное) значение терма $\mathfrak{a}$ и потому мы имеем на X t-членную операцию. Говорят, что эта операция изображается термом $\mathfrak{a}$.

При записи термов обычно пользуются теми или иными сокращениями. Например, если x—предметный f—одноместный функциональный символы, то терм $f(x)$ часто записывают в виде fx, xf или x^f. Далее, если x, y — предметные символы, а f — символ двуместной операции, то терм $f(x, y)$ обычно пишут в виде xfy. В частности, если рассматривают двуместные операции, обозначаемые символами $+, \cdot, -$, то всегда термы $+(\mathfrak{a}, \mathfrak{b}), \cdot(\mathfrak{a}, \mathfrak{b}), -(\mathfrak{a}, \mathfrak{b})$ сокращенно обозначают через $(\mathfrak{a})+(\mathfrak{b})$, $(\mathfrak{a})(\mathfrak{b})$, $(\mathfrak{a})-(\mathfrak{b})$. Если термы $\mathfrak{a}$, $\mathfrak{b}$ длины 1, то в указанной записи скобки опускают и пишут просто $\mathfrak{a}+\mathfrak{b}$, $\mathfrak{a}\mathfrak{b}$, $\mathfrak{a}-\mathfrak{b}$.

В дальнейшем всюду, где противное не оговорено явно, под числами понимаются натуральные числа 0, 1, 2, 3, . . . Цифры 0, 1, 2, . . ., 9 и их последовательности 10, 11, 102, . . . будут рассматриваться как предметные символы, имеющие фиксированные значения, равные соответственно натуральным числам нуль, один, . . ., десять, одиннадцать и т. д. Совокупность всех натуральных чисел будет обозначаться символом $\boldsymbol{N}$. Пустое множество будет обозначаться через $\varnothing$.

Частичные функции из $\boldsymbol{N} \times \ldots \times \boldsymbol{N} = \boldsymbol{N}^{(k)}$ в $\boldsymbol{N}$ будут называться *k-местными числовыми частичными функциями.* При этом для краткости иногда числовые функции будут называться просто функциями. Символы $+$, $\cdot$, $-$ будут рассматриваться как двуместные функциональные символы, фиксированными значениями которых являются обычные арифметические операции сложения, умножения и вычитания. В области натуральных чисел две первые операции всюду определенные, а операция вычитания частичная.

Далее нам будут часто встречаться числовые функции $\boldsymbol{s}^1$, $\boldsymbol{o}^n$, I^n_m, имеющие, по определению, следующие значения:

$$\boldsymbol{s}^1(x) = x + 1,$$

$$\boldsymbol{o}^n(x_1, \ldots, x_n) = 0,$$

$$I^n_m(x_1, \ldots, x_n) = x_m \quad (1 \leqslant m \leqslant n;\ n = 1, 2, \ldots).$$

Эти функции мы будем называть *простейшими.* Вместо $\boldsymbol{s}^1$, $\boldsymbol{o}^1$ мы будем обычно писать $\boldsymbol{s}$, $\boldsymbol{o}$.

Отметим, что терм, содержащий предметные символы x, y, z, представляет однозначно определенную операцию только в случае, если эти символы определенным образом упорядочены.

Изменяя их порядок, мы изменим и операцию. Например, терм

$$2x + y \tag{3}$$

представляет функцию, ставящую паре $\langle 1,3 \rangle$ в соответствие число 5, если переменная x считается первой, а переменная y — второй. Тот же терм (3) представляет функцию, относящую паре $\langle 1,3 \rangle$ число 7, если первой считается переменная y, а второй — переменная x.

здесь $b_0+b_1x_1+\ldots+b_nx_n = \underbrace{(1+\ldots+1)}_{b_0\text{-раз}}+\underbrace{(x_1+\ldots+x_1)}_{b_1\text{-раз}}+\ldots+\underbrace{(x_n+\ldots+x_n)}_{b_n\text{-раз}}$

Из рассмотренных примеров видно, что встречающиеся в термах предметные и функциональные символы могут быть разбиты на две категории: 1) символы, значения которых фиксированы, и 2) символы, значения которых не фиксируются. Первые называются *индивидуальными символами*, а вторые — *переменными.*

Пусть на некотором множестве X заданы операции $f_1, \ldots, f_s$ и фиксированы некоторые элементы $a_1, \ldots, a_r$ из X. Тогда каждая операция на X, представимая некоторым термом, записанным при помощи функциональных символов $f_1, \ldots, f_s$, предметных символов $a_1, \ldots, a_r$ и произвольных предметных переменных, называется термальной относительно данных операций и элементов. Например, если основное множество есть множество натуральных чисел $\boldsymbol{N}$, заданная операция есть $+$ и фиксировано число 1, то термальными будут все линейные функции вида

$$b_0+b_1x_1+\ldots+b_nx_n,$$

где b_i — положительные натуральные числа. Если число 1 не фиксируется, а задается лишь операция $+$, то термальными будут только однородные линейные функции вида $b_1x_1+\ldots+b_nx_n$, где b_i — снова положительные натуральные числа.

Часто понятие терма несколько расширяют. Именно, при заданных основных операциях $f_1, \ldots, f_s$ и элементах $a_1, \ldots, a_r$ n-местную операцию f называют термальной, если существует терм $\mathfrak{a}$, построенный из части символов $a_1, \ldots, a_r, f_1, \ldots, f_s$ и предметных переменных $x_{i_1}, \ldots, x_{i_k}$ $(1 \leqslant i_1 < \ldots < i_k \leqslant n)$, такой, что значение операции f в любой точке $\langle c_1, \ldots, c_n\rangle$ совпадает с значением терма при значениях переменных $x_{i_1}, \ldots, x_{i_k}$, равных соответственно $c_{i_1}, \ldots, c_{i_k}$. Это означает, например, что терм $x+y$ мы можем понимать либо как терм, представляющий двуместную функцию от x, y (старое определение), либо как терм, представляющий функцию от трех переменных x, y, z, где переменная z теперь входит в терм лишь фиктивно.

1.3. Алгебры. Алгеброй называется система, состоящая из некоторого непустого множества A и последовательности определенных на A операций $f_1^{n_1}, f_2^{n_2}, \ldots$

Множество A называется основным множеством алгебры, а операции $f_1^{n_1}$, $f_2^{n_2}$, ... называются ее *основными операциями*. Последовательность $\langle n_1, n_2, \ldots \rangle$ называется *типом* алгебры, а совокупность символов $f_1^{n_1}$, $f_2^{n_2}$, ..., обозначающих основные операции, называется *сигнатурой* алгебры. Алгебры называются однотипными, если типы их совпадают. *Классом* алгебр называется произвольная система однотипных алгебр. При исследовании классов алгебр соответствующие операции этих алгебр обозначаются обычно одними и теми же символами. Алгебра с основным множеством A и основными операциями f_1, f_2, ... обозначается через $\langle A; f_1, f_2, \ldots \rangle$. Если некоторые из операций f_1, f_2, ... частичные, то и алгебра называется частичной. Однотипные частичные алгебры $\mathfrak{A} = \langle A; f_1, f_2, \ldots \rangle$ и $\mathfrak{B} = \langle B; g_1, g_2, \ldots \rangle$ называются *изоморфными*, если существует одно-однозначное отображение φ множества A на множество B такое, что

$$f_i(a_1, \ldots, a_{n_i})^\varphi = g_i(a_1^\varphi, \ldots, a_{n_i}^\varphi) \qquad (i = 1, 2, \ldots) \quad (4)$$

для любых значений переменных $a_1, \ldots, a_{n_i}$, взятых в A (через a^φ обозначается образ элемента a при отображении φ).

Само отображение φ, обладающее свойством (4), называется *изоморфизмом* первой алгебры на вторую.

Если операции f_i, g_j частичные, то знак равенства в (4), как и всюду в дальнейшем, означает, что значение левой части, в том числе и «неопределенное значение», совпадает с значением правой части.

Пусть на множестве A определена какая-либо частичная операция f^n. Подмножество A_1 из A называется замкнутым относительно f^n, если для каждых элементов $a_1, \ldots, a_n$ из A_1, для которых значение $f^n(a_1, \ldots, a_n)$ определено, это значение принадлежит A_1.

Легко проверить, что пересечение любой системы подмножеств, замкнутых относительно операции f, либо пусто, либо замкнуто относительно f.

Рассмотрим теперь некоторое непустое множество A, на котором определены произвольные частичные операции f_1, f_2, ... Выделим в этом множестве какую-либо совокупность элементов D. Рассмотрим систему всех тех подмножеств из A, которые содержат D и замкнуты относительно

каждой из операций $f_1, f_2, \ldots$ Эта система непустая, так как само $A \supseteq D$ и замкнуто относительно операций $f_1, f_2, \ldots$ Пересечение D^* всех множеств этой системы замкнуто относительно рассматриваемых операций и содержит совокупность D. Это пересечение называется *подмножеством, порожденным в A элементами множества D с помощью операций* $f_1, f_2, \ldots$

Очевидно, подмножество D^* можно также определить и как наименьшее подмножество в A, содержащее D и замкнутое относительно рассматриваемых операций.

Несколько более ясное представление о строении множества D^* дает

Теорема 1. *Подмножество D^*, порожденное в A элементами совокупности D с помощью операций $f_1, f_2, \ldots$, состоит из элементов, являющихся значениями термов, записанных при помощи символов операций $f_1, f_2, \ldots$ и символов некоторых элементов D.*

Приведем доказательство. Пусть T — совокупность элементов, являющихся значениями указанных в теореме термов. Так как каждый элемент d из D является значением однобуквенного терма d, то T содержит D. Пусть $a_1, \ldots, a_{n_i}$ — элементы T, для которых $f_i(a_1, \ldots, a_{n_i})$ имеет значение a. По условию, $a_1, \ldots, a_{n_i}$ являются значениями некоторых термов $\mathfrak{a}_1, \ldots, \mathfrak{a}_{n_i}$ указанного выше вида. Но тогда a есть значение терма $f_i(\mathfrak{a}_1, \ldots, \mathfrak{a}_{n_i})$, также имеющего указанный выше вид. Следовательно, $a \in T$ и, значит, T замкнута относительно операций $f_1, f_2, \ldots$ Из замкнутости T и условия $D \subseteq T$ следует, что $D^* \subseteq T$.

Остается доказать, что T содержится в D^*. Для этого достаточно доказать, что каждое подмножество $B \subseteq A$, замкнутое относительно операций $f_1, f_2, \ldots$ и содержащее совокупность D, содержит значение каждого терма указанного в теореме вида. Для термов длины 1 это утверждение очевидно. Если же терм имеет вид $f_i(\mathfrak{a}_1, \ldots, \mathfrak{a}_{n_i})$ и значения термов $\mathfrak{a}_1, \ldots, \mathfrak{a}_{n_i}$ содержатся в B, то из замкнутости множества B следует, что значение терма $f_i(\mathfrak{a}_1, \ldots$ $\ldots, \mathfrak{a}_{n_i})$ также содержится в B. В силу индукции наше утверждение можно считать доказанным, а вместе с ним доказана и теорема.

Рассмотрим некоторую частичную алгебру $\mathfrak{A} = \langle A; f_1, f_2, \ldots \rangle$. Берем произвольное непустое подмножество A_1 из A и определяем на A_1 операцию $\bar{f}_i$, полагая для произвольных $a_1, \ldots, a_{n_i}$ из A_1 значение $\bar{f}_i(a_1, \ldots, a_{n_i})$ равным значению $f_i(a_1, \ldots, a_{n_i})$, если последнее входит в A_1, и полагая $\bar{f}_i(a_1, \ldots, a_{n_i})$ неопределенным, если значение $f_i(a_1, \ldots, a_{n_i})$ не входит в A_1 или не определено. Алгебра $\mathfrak{A}_1 = \langle A_1; \bar{f}_1, \bar{f}_2, \ldots \rangle$ называется частичной подалгеброй алгебры $\mathfrak{A}$. Для операций $\bar{f}_1, \bar{f}_2, \ldots$ обычно употребляются те же символы $f_1, f_2, \ldots$, что и для операций самой алгебры $\mathfrak{A}$, и вместо «подалгебра $\mathfrak{A}_1 = \langle A_1; f_1, f_2, \ldots \rangle$» говорят кратко: «подалгебра A_1 алгебры A». Частичная подалгебра $\mathfrak{A}_1$ называется замкнутой, если множество A_1 замкнуто относительно операций $f_1, f_2, \ldots$ В силу сказанного выше, пересечение замкнутых частичных подалгебр любой частичной алгебры либо пусто, либо является замкнутой частичной подалгеброй.

Частичная алгебра называется *алгеброй*, если все ее основные операции всюду определенные. Аналогично, частичная подалгебра называется просто *подалгеброй*, если она есть алгебра. Легко проверить, что частичная подалгебра произвольной алгебры тогда и только тогда является просто подалгеброй, когда она замкнутая. Таким образом, каждое множество, замкнутое относительно основных операций алгебры, вместе с этими операциями образует подалгебру этой алгебры.

Говорят, что совокупность D элементов частичной алгебры $\mathfrak{A} = \langle A; f_1, f_2, \ldots \rangle$ *порождает* частичную подалгебру $\mathfrak{A}_1 = \langle A_1; f_1, f_2, \ldots \rangle$, если D порождает множество A_1 с помощью основных операций $f_1, f_2, \ldots$ в A. Элементы множества D называются *порождающими* частичной алгебры $\mathfrak{A}$, если порожденная совокупностью D частичная подалгебра совпадает с $\mathfrak{A}$.

Принимая во внимание теорему 1, видим, что *совокупность D элементов частичной алгебры $\mathfrak{A}$ тогда и только тогда является порождающей совокупностью этой алгебры, когда каждый элемент $\mathfrak{A}$ есть значение подходящего терма, образованного из символов основных операций $\mathfrak{A}$ и символов элементов D*. Иначе говоря, элементы D тогда и только

тогда порождают алгебру $\mathfrak{A}$, когда каждый элемент $\mathfrak{A}$ можно получить из элементов D, применяя конечное число раз основные операции алгебры.

Частичная алгебра называется *конечно порожденной*, если она порождается каким-либо конечным множеством своих элементов.

Например, алгебра $\langle \boldsymbol{N}; +\rangle$ конечно порожденная. В качестве порождающего множества можно взять любое множество чисел, содержащее 0, 1.

Напротив, алгебра $\langle \boldsymbol{N}; \times\rangle$ не является конечно порожденной. Действительно, беря какое-нибудь конечное множество чисел $a_1, \ldots, a_r$, мы с помощью операции умножения можем получить из них числа, делящиеся лишь на те простые числа, на которые делится хотя бы одно из заданных чисел $a_1, \ldots, a_r$. Так как простых чисел бесконечно много, то все натуральные числа получить умножениями из чисел $a_1, \ldots, a_r$ нельзя.

1.4. Кодирование. Теория алгоритмов имеет дело со словами в конечном алфавите. Однако существуют математические объекты, играющие основную роль, которые нельзя без натяжки непосредственно рассматривать как слова в некотором алфавите. К таким объектам, например, можно отнести рациональные числа, алгебраические числа, функции и многое другое. Тем не менее ряд таких объектов может быть охарактеризован конечным числом целочисленных параметров или просто словом в подходящем конечном алфавите. Например, возьмем алфавит, состоящий лишь из знака |. Словами в этом алфавите являются последовательности «палочек»: |, ||, |||, . . . Сопоставим с каждым словом ||. . . |, содержащим $n+1$ «палочек», натуральное число n. Говорят, что указанное слово изображает число n или является *кодом* числа n.

Аналогичным образом, вместо алфавита, состоящего из одной буквы |, можно рассмотреть алфавит, состоящий из двух букв 0, 1, и в качестве кодированной записи натурального числа n брать его двоичную запись. Например, кодом числа пять в новом алфавите будет слово 101.

При двоичном кодировании мы встречаемся с той особенностью, что не каждое слово является кодом какого-нибудь натурального числа. Например, слово 01 не является кодом никакого натурального числа.

Рассмотрим еще обычное кодирование положительных рациональных чисел. Вводим алфавит, состоящий из прямой черты | и косой черты /. Слово вида ||. . . |/| . . . ||, состоящее из $m+1$ знаков |, косой черты / и последующих $n+2$ знаков |, условимся называть кодом рационального числа $\frac{m}{n+1}$. Ясно, что теперь не только не каждое слово является кодом рационального числа, но каждое положительное рациональное число имеет бесконечно много различных кодов. Например, число $^1/_2$ имеет коды ||/|||, |||/||||| и т. д.

Во всех указанных случаях было выполнено следующее существенное требование: по заданному коду закодированный объект восстанавливался однозначно. Это требование обычно сохраняется и в общем определении кодирования.

Пусть $\mathfrak{M}$ — совокупность некоторых объектов, A — какой-нибудь конечный алфавит. Закодировать совокупность $\mathfrak{M}$ в алфавите A — это значит задать однозначное отображение $\varkappa$ некоторого множества $\mathfrak{A}_\varkappa$ слов в алфавите A на совокупность $\mathfrak{M}$. Слово $\mathfrak{a}$ из множества $\mathfrak{A}_\varkappa$ называется *кодом* (иногда — *именем*) объекта $\varkappa(\mathfrak{a})$ при кодировании $\varkappa$.

В этом определении не требуется, чтобы отображение $\varkappa$ было одно-однозначным. Поэтому может случиться, что один и тот же объект будет иметь несколько различных кодов.

Отображение $\varkappa$ и множество $\mathfrak{A}_\varkappa$ могут быть очень сложными, но обычно рассматриваются и называются кодированиями лишь несложные отображения. Рассмотрим несколько наиболее часто встречающихся кодирований.

а) Кодирование слов в многобуквенном алфавите. Пусть алфавит A состоит из букв 0, 1, а алфавит B состоит из букв $b_1, b_2, \ldots, b_n, \ldots$ Рассмотрим совокупность $\mathfrak{A}_\varkappa$ тех слов в алфавите A, которые начинаются с 1, оканчиваются буквой 0 и не содержат подслов вида 00. Обозначим через 1^n слово $111\ldots1$ длины n. Каждое слово из $\mathfrak{A}_\varkappa$ однозначно записывается в виде $1^{i_1}01^{i_2}0\ldots1^{i_k}0$. Этому слову ставим в соответствие слово $b_{i_1}b_{i_2}\ldots b_{i_k}$ алфавита B. Ясно, что полученное кодирование одно-однозначное.

В рассмотренном случае алфавит B бесконечен. Если он конечен, то в качестве кода можно взять то же отображение $\varkappa$. Изменится лишь множество $\mathfrak{A}_\varkappa$.

б) Бесконечные алфавиты. Согласно основному определению, алфавитом называется *конечное* множество символов. Однако фактически очень часто удобнее рассматривать бесконечные алфавиты, состоящие из конечного числа букв, снабженных той или иной системой индексов, принимающих натуральные значения 0, 1, 2, ... Например, пусть алфавит B состоит из символов $a, b, \ldots$ $\ldots, c, f_i, g_i^j$ $(i, j = 0, 1, 2, \ldots)$. Этот алфавит формально бесконечен. Рассмотрим алфавит A, состоящий из символов $a, b, \ldots, c, f, g$ и новых символов α, β. Условимся символы f_i и g_i^j кодировать в языке A словами $f\alpha \ldots \alpha$ и, соответственно, $g\alpha \ldots \alpha\beta \ldots \beta$, где символ α входит i раз, а символ β входит j раз. Символы $a, b, \ldots, c$ пусть кодируются ими самими. Обозначим через $\{x\}$ $(x \in B)$ кодовое значение символа x в алфавите A. Если $\mathfrak{a}$ — слово в алфавите B, то, подставляя в $\mathfrak{a}$ вместо символов их кодовые значения, получим некоторое слово в алфавите A, которое обозначим через $\{\mathfrak{a}\}$. Ясно, что по коду $\{\mathfrak{a}\}$ однозначно восстанавливается и слово $\mathfrak{a}$. Поэтому, вместо того чтобы рассматривать произвольные слова $\mathfrak{a}, \mathfrak{b}, \ldots$ в бесконечном алфавите B, можно рассматривать их кодовые значения $\{\mathfrak{a}\}, \{\mathfrak{b}\}, \ldots$ в конечном алфавите A и тем самым избежать рассмотрения бесконечных алфавитов.

в) Последовательности слов. Пусть A — некоторый алфавит. Обозначим через B алфавит, получающийся добавлением к A нового символа α $(\alpha \notin A)$. В дальнейшем наряду с отдельными словами $\mathfrak{a}, \mathfrak{b}, \ldots$ в алфавите A придется часто рассматривать пары слов $\langle \mathfrak{a}, \mathfrak{b} \rangle$, тройки слов $\langle \mathfrak{a}, \mathfrak{b}, \mathfrak{c} \rangle$ и т. д. Однако пару слов $\langle \mathfrak{a}, \mathfrak{b} \rangle$ в алфавите A можно закодировать словом $\mathfrak{a}\alpha\mathfrak{b}$ в алфавите B, тройку $\langle \mathfrak{a}, \mathfrak{b}, \mathfrak{c} \rangle$ слов из A можно закодировать словом $\mathfrak{a}\alpha\mathfrak{b}\alpha\mathfrak{c}$ в алфавите B и т. д.

Примеры и задачи

1. Пусть N — совокупность натуральных чисел, $+$ — обычная арифметическая операция сложения чисел. Показать, что в алгебре $\mathfrak{A} = \langle N; + \rangle$ число 2 порождает множество всех положительных четных чисел, число 1 порождает множество всех положительных чисел, а числа 0 и 1 порождают всю алгебру $\mathfrak{A}$.

2. Показать, что все подалгебры алгебры $\mathfrak{B} = \langle N; +, - \rangle$ исчерпываются следующими: а) подалгебра (0), порожденная

числом 0 и состоящая только из него; б) подалгебра (1), порожденная числом 1 и совпадающая с самой алгеброй $\mathfrak{B}$; в) подалгебра (a), порожденная произвольным числом $a > 1$, состоящая из чисел $0, a, 2a, 3a, \ldots$

3. Показать, что каждая подалгебра алгебры $\mathfrak{A} = \langle N; + \rangle$ состоит из всевозможных кратных подходящего числа a, возможно, за исключением конечного числа таковых.

4. Рассмотрим алгебру $\mathfrak{C} = \langle N; +, \cdot, - \rangle$, где операция — рассматривается как частичная двуместная операция. Значение терма $x - y$ считается равным обычной разности для $x \geqslant y$ и неопределенным для $x < y$. Какие функции представляются термами $0 \cdot (x - (x + 1))$, $(x - y) + y$, $(x + y) - y$? Показать, что функция 2^x не может быть представлена термом в алгебре $\mathfrak{C}$ (то есть термом, записанным в алфавите $x, +, \cdot, -$).

§ 2. ОСНОВНЫЕ ВЫЧИСЛИМЫЕ ОПЕРАТОРЫ

Операции над числовыми функциями далее называются операторами. В настоящем параграфе определяется ряд операторов, обладающих тем свойством, что, применяя их к функциям, вычислимым в интуитивном смысле (см. введение), мы получим функции, также заведомо вычислимые в интуитивном смысле. Частичные функции, которые можно получить при помощи этих операторов из простейших функций $\mathbf{s}(x) = x + 1$, $\mathbf{o}(x) = 0$, $I^n_m(x_1, \ldots, x_n) = x_m$, называются частично рекурсивными. Основная гипотеза Чёрча состоит в том, что класс частично рекурсивных функций совпадает с классом функций, допускающих машинное или алгоритмическое вычисление.

2.1. Суперпозиции частичных функций. Пусть заданы n каких-либо частичных функций $f_1, \ldots, f_n$ от одного и того же числа m переменных, определенных на каком-то множестве A со значениями в множестве B, и пусть на множестве B определена частичная функция f от n переменных, значения которой принадлежат некоторому третьему множеству C. Вводим теперь частичную функцию g от m переменных, определенную на A со значениями в C, полагая по определению

$$g(x_1, \ldots, x_m) = f(f_1(x_1, \ldots, x_m), \ldots, f_n(x_1, \ldots, x_m))$$

для произвольных $x_1, \ldots, x_m$ из A. Говорят, что функция g получается операцией суперпозиции или подстановки

из функций f, $f_1, \ldots, f_n$. Операция подстановки будет далее обозначаться символом S^{n+1}. Индекс сверху означает число функций. В качестве множеств A, B, C ниже почти всюду будет браться множество натуральных чисел N. Условимся через $\mathfrak{F}^n$ обозначать совокупность всех частичных числовых функций от n переменных. Оператор S^{n+1} является всюду определенной функцией из $\mathfrak{F}^n \times \mathfrak{F}^m \times \ldots \times \mathfrak{F}^m$ в $\mathfrak{F}^m$. Если же рассматривать множество $\mathfrak{F}$ всех частичных числовых функций от произвольного числа переменных, то оператор S^{n+1} можно рассматривать как частичную функцию от $n+1$ переменной, определенную на $\mathfrak{F}$ со значениями в $\mathfrak{F}$. При этом терм $S^{n+1}(f, f_1, \ldots, f_n)$ имеет определенное значение (равное частичной m-местной функции) тогда и только тогда, когда значением переменной f будет n-местная функция, а значениями символов $f_1, \ldots, f_n$ будут частичные функции от одного и того же числа переменных. Например, с последней точки зрения значение терма $S^3(I_1^2, I_1^3, I_2^2)$ не определено, а значение терма $S^3(I_1^2, I_1^3, I_2^3)$ равно I_1^3, где I_1^2, I_1^3, I_2^3 — символы простейших функций, определенных в п. 1.2.

В соответствии с п. 1.3 система $\mathfrak{P}$, состоящая из совокупности функций $\mathfrak{F}$ и определенных на ней частичных операций S^2, $S^3, \ldots$, будет частичной алгеброй.

Пусть $f_1^{n_1}, \ldots, f_s^{n_s}$ — какие-нибудь частичные числовые функции соответственно от $n_1, \ldots, n_s$ аргументов ($n_1, \ldots, n_s$ произвольны). Частичные функции, которые можно получить операциями подстановок из $f_1^{n_1}, \ldots, f_s^{n_s}$ и функций I_m^n ($m, n = 1, 2, \ldots$), называются элементарными относительно $f_1^{n_1}, \ldots, f_s^{n_s}$. Совокупность их будет в алгебре $\mathfrak{P}$ подалгеброй, порожденной элементами $f_1^{n_1}, \ldots, f_s^{n_s}, I_m^n$. Теорема 1 из п. 1.3 показывает, что относительно элементарными будут те и только те частичные функции, которые можно записать в виде термов, содержащих функциональные символы S^2, $S^3, \ldots$ операций подстановки, символы $f_1^{n_1}, \ldots, f_s^{n_s}$ заданных и символы I_m^n простейших функций. Однако для относительно элементарных функций возможна и еще более простая запись.

Теорема 1. *Для того чтобы некоторая n-местная частичная числовая функция f была элементарной относительно частичных функций $f_1, \ldots, f_s$, необходимо и достаточно, чтобы f можно было представить в виде терма,*

записанного с помощью функциональных символов $f_1, f_2, \ldots$ $\ldots, f_s$ *и некоторых предметных переменных* $x_1, \ldots, x_n$, *часть из которых может входить в терм и фиктивно.*

Итак, утверждается, что для относительно элементарной частичной функции f существует две записи: 1) в виде значения терма, функциональными символами которого служат символы $\boldsymbol{S}^i$ операций подстановки, а предметными символами — символы $f_1, \ldots, f_s, I^n_m$ (термы этого вида мы будем называть операторными); 2) в виде терма (а не значения терма), составленного из функциональных символов $f_1, \ldots, f_s$ и некоторых предметных переменных (может быть, фиктивных). Эту запись мы будем называть *термальной записью.* Отметим еще раз, что в первом случае функция f *есть значение терма*, а во втором—функция f *представляется термом.*

Для доказательства теоремы надо показать, что от термальной записи можно переходить к операторной и наоборот. Пусть f — значение операторного терма $\mathfrak{a}$. Если длина $\mathfrak{a}$ равна 1, то $\mathfrak{a}$ есть либо f_i, либо I^n_j. В первом случае термальным представлением для f является терм $f_i(x_{i_1}, \ldots$ $\ldots, x_{i_n})$, а во втором случае — терм x_j. Если длина $\mathfrak{a}$ больше 1, то $\mathfrak{a}$ имеет вид $\boldsymbol{S}^{k+1}(\mathfrak{b}, \mathfrak{b}_1, \ldots, \mathfrak{b}_k)$, где $\mathfrak{b}, \mathfrak{b}_1, \ldots, \mathfrak{b}_k$ — операторные термы меньшей длины. По индукции предполагаем, что термальные представления

$$\mathfrak{c} = \mathfrak{c}(x_1, \ldots, x_k), \quad \mathfrak{c}_i = \mathfrak{c}_i(x_1, \ldots, x_n) \ (i = 1, \ldots, k)$$

для значений термов $\mathfrak{b}, \mathfrak{b}_1, \ldots, \mathfrak{b}_k$ нам известны. Тогда

$$\mathfrak{c}(\mathfrak{c}_1(x_1, \ldots, x_n), \ldots, \mathfrak{c}_k(x_1, \ldots, x_n))$$

будет искомым термальным представлением для заданного операторного терма $\mathfrak{a}$.

Обратно, пусть n-местная частичная функция f задана своим термальным представлением $\mathfrak{a}(x_1, \ldots, x_n)$, где переменные $x_{i_1}, \ldots, x_{i_k}$ фактически входят в терм, а остальные из переменных $x_1, \ldots, x_n$ рассматриваются как фиктивные. Если длина терма $\mathfrak{a}$ равна 1, то $\mathfrak{a}$ имеет вид x_j и, значит, f равна значению операторного терма I^n_j. Если же длина терма $\mathfrak{a}$ больше 1, то $\mathfrak{a}$ имеет вид $f_i(\mathfrak{a}_1, \ldots, \mathfrak{a}_{n_i})$, где $\mathfrak{a}_1, \ldots, \mathfrak{a}_{n_i}$ — термы меньшей длины. По индукции

предполагаем, что для каждой n-местной частичной функции от $x_1, \ldots, x_n$, представляемой термом $\mathfrak{a}_j$, найдено ее выражение в виде операторного терма $\mathfrak{c}_j$ ($j = 1, \ldots, n_i$). Тогда ясно, что функция f будет значением операторного терма $\boldsymbol{S}^{n_i+1}(f_i, \mathfrak{c}_1, \ldots, \mathfrak{c}_{n_i})$.

Например, пусть заданы обычные функции $+$ и $\times$. Функция от x_1, x_2, x_3, имеющая термальное представление $x_1 \cdot x_2 + x_3$, является значением операторного терма

$$\boldsymbol{S}^3(+, \boldsymbol{S}^3(\times, I_1^3, I_2^3), I_3^3).$$

2.2. Оператор примитивной рекурсии. Пусть заданы какие-нибудь числовые частичные функции: n-местная g и $(n+2)$-местная h. Говорят, что $(n+1)$-местная частичная функция f *возникает из функций g и h примитивной рекурсией*, если для всех натуральных значений $x_1, \ldots, x_n, y$ имеем

$$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n), \tag{1}$$

$$f(x_1, \ldots, x_n, y+1) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y)). \tag{2}$$

Это определение мы будем применять и для $n = 0$, говоря, что одноместная частичная функция f возникает примитивной рекурсией из постоянной одноместной функции, равной числу a, и двуместной частичной функции h, если

$$f(0) = a, \tag{3}$$

$$f(x+1) = h(x, f(x)). \tag{4}$$

Встает вопрос, для каждых ли частичных функций g, h от n и $n+2$ переменных существует частичная функция f от $n+1$ переменной, удовлетворяющая условиям (1), (2) или соответственно (3), (4), и будет ли такая частичная функция единственной? Так как область определения функций есть множество всех натуральных чисел, то ответ на оба вопроса, очевидно, положительный. Если функция f существует, то из (1) и (2) последовательно находим

$$\begin{gathered} f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n), \\ f(x_1, \ldots, x_n, 1) = h(x_1, \ldots, x_n, 0, g(x_1, \ldots, x_n)), \qquad (5) \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ f(x_1, \ldots, x_n, m+1) = h(x_1, \ldots, x_n, m, f(x_1, \ldots, x_n, m)) \end{gathered}$$

и потому f определена однозначно. Из соотношений (5), в частности, видно, что если для некоторых $x_1, \ldots, x_n, t$ значение $f(x_1, \ldots, x_n, t)$ неопределенное, то неопределенными будут и значения $f(x_1, \ldots, x_n, y)$ для всех $y \geqslant t$.

Чтобы при заданных частичных функциях g, h найти функцию f, возникающую из них примитивной рекурсией, достаточно равенства (5) принять в качестве равенств, определяющих значения функции f. Таким образом, для любых частичных n-местной функции g и $(n+2)$-местной функции h $(n = 0, 1, 2, \ldots)$ существует одна и только одна частичная $(n+1)$-местная функция f, возникающая из g и h примитивной рекурсией. Символически пишут

$$f = \boldsymbol{R}(g, h)$$

и рассматривают $\boldsymbol{R}$ как символ двуместной частичной операции, определенной на множестве $\mathfrak{F}$ всех частичных функций. Из соотношений (5) вытекает, что если функции g и h всюду определенные, то и функция f всюду определенная.

Из соотношений (5) видно также и следующее фундаментальное для нас обстоятельство: если мы каким-то образом «умеем» находить значения функций g, h, то значения функции f можно вычислять при помощи процедуры вполне «механического» характера. Именно, для нахождения значения $f(a_1, \ldots, a_n, m+1)$ достаточно последовательно найти числа

$$b_0 = g(a_1, \ldots, a_n),$$

$$b_1 = h(a_1, \ldots, a_n, 0, b_0),$$

$$b_2 = h(a_1, \ldots, a_n, 1, b_1),$$

$$\cdots\cdots\cdots\cdots\cdots$$

$$b_{m+1} = h(a_1, \ldots, a_n, m, b_m).$$

Полученное на $(m+1)$-м «шаге» число b_{m+1} и будет искомым значением функции f в точке $\langle a_1, \ldots, a_n, m+1\rangle$. Изложенный процесс вычисления $f(a_1, \ldots, a_n, m+1)$ будет продолжаться неограниченно только в том случае, когда неограниченным окажется процесс вычисления

одного из выражений

$$g(a_1, \ldots, a_n), \quad h(a_1, \ldots, a_n, 0, b_0), \ldots$$
$$\ldots, h(a_1, \ldots, a_n, m, b_m),$$

т. е. когда хотя бы одно из этих выражений будет иметь неопределенное значение. Но тогда, в соответствии с определением функции f, значение $f(a_1, \ldots, a_n, m+1)$ будет неопределенным.

Теперь мы введем одно из главных понятий теории рекурсивных функций. Пусть задана система $\mathfrak{S}$ каких-то частичных функций. Частичная функция f называется *примитивно рекурсивной относительно* $\mathfrak{S}$, если f можно получить из функций системы $\mathfrak{S}$ и простейших функций $\boldsymbol{s}$, $\boldsymbol{o}$, I^n_m конечным числом операций подстановки и примитивной рекурсии.

Функция f называется просто *примитивно рекурсивной*, если ее можно получить конечным числом операций подстановки и примитивной рекурсии, исходя лишь из простейших функций $\boldsymbol{s}$, $\boldsymbol{o}$, I^n_m.

Операции подстановки и примитивной рекурсии, будучи применены к всюду определенным функциям, дают в результате снова всюду определенные функции. Поэтому, если задана система $\mathfrak{S}$ всюду определенных функций, то примитивно рекурсивные относительно $\mathfrak{S}$ функции всюду определенные. В частности, *все примитивно рекурсивные функции всюду определенные.*

Ясно, что частичная функция f, примитивно рекурсивная относительно какой-нибудь системы $\mathfrak{S}$ частичных функций, примитивно рекурсивна и относительно любой более широкой системы. Примитивно рекурсивные функции можно рассматривать как примитивно рекурсивные относительно пустой системы $\mathfrak{S}$. Поэтому примитивно рекурсивные функции — это функции, примитивно рекурсивные относительно любой системы функций.

Наконец, из определения также непосредственно вытекает, что *операции подстановки и примитивной рекурсии, примененные к частичным функциям, примитивно рекурсивным относительно какой-нибудь системы $\mathfrak{S}$, дают в результате снова функции, примитивно рекурсивные относительно $\mathfrak{S}$.*

Подробно свойства примитивно рекурсивных функций будут изучены в § 3, а здесь мы в качестве примера докажем лишь примитивную рекурсивность самых простых арифметических функций.

Согласно определению, одноместные функции $\mathbf{s}(x) = x + 1$, $\mathbf{o}^1(x) = 0$, $I_1^1(x) = x$ и многоместные функции

$$I_m^n(x_1, \ldots, x_n) = x_m \quad (m, n = 1, 2, \ldots)$$

примитивно рекурсивны.

Для n-местной функции $\mathbf{o}^n(x_1, \ldots, x_n) = 0$ имеем представление

$$\mathbf{o}^n = \mathbf{S}^2(\mathbf{o}^1, I_1^n)$$

и потому функция $\mathbf{o}^n$ примитивно рекурсивная. Произвольная n-местная постоянная функция $f^n = a$ допускает представление в виде терма

$$\mathbf{s}(\mathbf{s}(\ldots \mathbf{s}(\mathbf{o}^n(x_1, \ldots, x_n)) \ldots)),$$

записанного при помощи символов примитивно рекурсивных функций $\mathbf{s}$, $\mathbf{o}^n$ и предметных переменных. Поэтому все постоянные функции примитивно рекурсивны.

Двуместная функция $f(x, y) = x + y$ удовлетворяет соотношениям

$$x + 0 = x = I_1^1(x),$$
$$x + (y + 1) = (x + y) + 1 = \mathbf{s}(x + y).$$

Следовательно, функция $x + y$ возникает из примитивно рекурсивных функций I_1^1, $h(x, y, z) = z + 1$ операцией примитивной рекурсии и потому функция $x + y$ примитивно рекурсивная.

Двуместная функция xy удовлетворяет схеме примитивной рекурсии

$$x \cdot 0 = \mathbf{o}(x),$$
$$x(y + 1) = xy + x$$

с начальными примитивно рекурсивными функциями (см. (1), (2))

$$g(x) = \mathbf{o}(x), \quad h(x, y, z) = z + x.$$

Поэтому функция xy примитивно рекурсивна.

Рассмотрим функцию x^y, причем будем считать, что $x^0 = 1$. Соотношения

$$x^0 = 1,$$
$$x^{y+1} = x^y x$$

представляют собой примитивно рекурсивную схему с начальными примитивно рекурсивными функциями

$$g(x) = 1, \quad h(x, y, z) = z \cdot x.$$

Поэтому функция x^y также примитивно рекурсивна.

В анализе иногда встречается функция $\operatorname{sg} x$ (*сигнум* или *знак* x), равная $+1$ для положительных вещественных значений символа x, -1 для отрицательных x и 0 для $x = 0$. Мы будем рассматривать эту функцию лишь для натуральных значений x, для которых она может быть определена таблицей

$$\operatorname{sg} x = \begin{cases} 0, & \text{если} \quad x = 0, \\ 1, & \text{если} \quad x > 0. \end{cases}$$

Введем еще функцию $\overline{\operatorname{sg}}$, определяемую таблицей

$$\overline{\operatorname{sg}}\, x = \begin{cases} 1, & \text{если} \quad x = 0, \\ 0, & \text{если} \quad x > 0, \end{cases}$$

и совпадающую с разностью $1 - \operatorname{sg} x$.

Функции sg и $\overline{\operatorname{sg}}$ удовлетворяют примитивно рекурсивным схемам:

$$\operatorname{sg} 0 = 0, \qquad \overline{\operatorname{sg}}\, 0 = 1,$$
$$\operatorname{sg}(x+1) = 1, \qquad \overline{\operatorname{sg}}(x+1) = 0,$$

имеющим вид (3), (4). Поэтому функции sg и $\overline{\operatorname{sg}}$ примитивно рекурсивные.

В области натуральных чисел разность $x - y$ естественно считать *частичной* двуместной функцией от x, y, определенной лишь для $x \geqslant y$, так как отрицательные числа не входят в рассматриваемую область. Но примитивно рекурсивные функции всюду определенные. Поэтому в теории примитивно рекурсивных функций вместо обычной разности вводят усеченную разность, обозначаемую

символом $\dot{-}$ и определяемую таблицей

$$x \dot{-} y = \begin{cases} x - y, & \text{если} \quad x \geqslant y, \\ 0, & \text{если} \quad x < y. \end{cases} \tag{6}$$

В отличие от простой разности усеченная разность является в области натуральных чисел всюду определенной, и в то же время она очень просто связана с обычной разностью. Так, например, согласно (6)

$$5 \dot{-} 3 = 2, \quad 3 \dot{-} 5 = 0, \quad (x \dot{-} y) \dot{-} z = x \dot{-} (y + z).$$

Функция $x \dot{-} 1$ удовлетворяет примитивно рекурсивной схеме

$$0 \dot{-} 1 = 0,$$
$$(x + 1) \dot{-} 1 = x$$

с примитивно рекурсивными начальными функциями $\boldsymbol{O}^1$ и I_1^2. Поэтому функция $x \dot{-} 1$ примитивно рекурсивная.

С другой стороны, из (6) следует, что для любых x, y

$$x \dot{-} 0 = x,$$
$$x \dot{-} (y + 1) = (x \dot{-} y) \dot{-} 1.$$

Эти тождества показывают, что двуместная функция $x \dot{-} y$ возникает примитивной рекурсией из функций I_1^1 и $h(x, y, z) = z \dot{-} 1$. Обе последние функции примитивно рекурсивны. Поэтому и функция $x \dot{-} y$ примитивно рекурсивная.

Наконец, из примитивной рекурсивности функций $+$ и $\dot{-}$ вытекает примитивная рекурсивность функции

$$|x - y| = (x \dot{-} y) + (y \dot{-} x).$$

На этом мы пока прервем изучение свойств примитивно рекурсивных функций. Оно будет продолжено в § 3, а сейчас мы еще раз обратимся к определению относительно примитивно рекурсивных функций.

В настоящем и предыдущем пунктах на множестве $\mathfrak{F}$ всех частичных функций с произвольным числом аргументов были определены частичные операции суперпозиции $\boldsymbol{S}^i$ $(i = 2, 3, \ldots)$ и примитивной рекурсии $\boldsymbol{R}$. Согласно п. 1.3 система

$$\mathfrak{B} = \langle \mathfrak{F}; \boldsymbol{R}, \boldsymbol{S}^2, \boldsymbol{S}^3, \ldots \rangle$$

является частичной алгеброй. Из приведенных выше определений непосредственно вытекает, что совокупность $\mathfrak{F}_{\text{пр}}$

всех примитивно рекурсивных функций является подалгеброй алгебры $\mathfrak{B}$, порожденной системой функций $\boldsymbol{o}$, $\boldsymbol{s}$, I^n_m (m, $n = 1, 2, \ldots$). Аналогично, если $\mathfrak{S}$ — система какихлибо функций из $\mathfrak{F}$, то совокупность всех функций, примитивно рекурсивных относительно $\mathfrak{S}$, является подалгеброй алгебры $\mathfrak{B}$, порожденной функциями системы $\mathfrak{S}$ и простейшими функциями $\boldsymbol{o}$, $\boldsymbol{s}$, I^n_m. В силу теоремы 1 из п. 1.3 можно утверждать, что *примитивно рекурсивными являются те и только те функции, которые являются значениями термов, записываемых с помощью индивидуальных предметных символов* $\boldsymbol{o}$, $\boldsymbol{s}$, I^n_m *и функциональных символов* $\boldsymbol{R}$, $\boldsymbol{S}^2$, $\boldsymbol{S}^3$, ...

Аналогично, те и только те функции примитивно рекурсивны относительно $\mathfrak{S}$, которые являются значениями термов, записываемых с помощью индивидуальных предметных символов $\boldsymbol{o}$, $\boldsymbol{s}$, I^n_m, символов функций из $\mathfrak{S}$ и функциональных символов $\boldsymbol{R}$, $\boldsymbol{S}^2$, $\boldsymbol{S}^3$.

Термы этих видов мы, как и в п. 2.1, будем называть операторными. С формальной точки зрения операторные термы — это слова в алфавите, образованном символами $\boldsymbol{R}$, $\boldsymbol{S}^i$, $\boldsymbol{o}$, $\boldsymbol{s}$, I^n_m и символами, которые обозначают функции из системы $\mathfrak{S}$. Как правило, система $\mathfrak{S}$ предполагается конечной, но даже и в этом случае алфавит получается бесконечным, так как он включает бесконечное число символов $\boldsymbol{S}^2$, $\boldsymbol{S}^3$, . . . Однако эту сложность легко устранить, если воспользоваться надлежащим кодированием, указанным, например, в п. 1.4. Тогда каждая примитивно рекурсивная функция будет значением подходящего слова в фиксированном конечном алфавите.

Запись примитивно рекурсивной функции в виде операторного терма будет рассматриваться далее как стандартное (конструктивное) задание этой функции. Как правило, доказательство примитивной рекурсивности некоторой функции будет состоять в указании пути для построения операторного терма, значение которого равно данной функции. Например, приведенное выше доказательство примитивной рекурсивности усеченной разности дает следующие термальные представления:

$$x \dot{-} 1 = \boldsymbol{R}(\boldsymbol{o}, I^1_1)(x),$$

$$\dot{-} = \boldsymbol{R}(I^1_1, I^3_3 \dot{-} 1) = \boldsymbol{R}(I^1_1, \boldsymbol{S}^2(\boldsymbol{R}(\boldsymbol{o}, I^2_1), I^3_3)).$$

Отметим сразу же, что функция $\dot{-}$ допускает и бесконечно много других термальных представлений, так как значение терма не изменится, если к нему приписать, например, слово $+\boldsymbol{o}$.

2.3. Операция минимизации. Рассмотрим какую-нибудь n-местную ($n \geqslant 1$) частичную числовую функцию f. Допустим, что существует какой-либо «механизм» для вычисления значений функции f, причем значение функции f тогда и только тогда неопределенное, когда этот механизм работает бесконечно, не выдавая никакого определенного результата. Фиксируем какие-нибудь значения $x_1, \ldots, x_{n-1}$ для первых $n - 1$ аргументов функции f и рассмотрим уравнение

$$f(x_1, \ldots, x_{n-1}, y) = x_n. \tag{1}$$

Чтобы найти решение y (натуральное) этого уравнения, будем вычислять при помощи указанного выше «механизма» последовательно значения $f(x_1, \ldots, x_{n-1}, y)$ для $y = 0, 1, 2, \ldots$ Наименьшее значение a, для которого получится

$$f(x_1, \ldots, x_{n-1}, a) = x_n,$$

обозначим через

$$\mu_y(f(x_1, \ldots, x_{n-1}, y) = x_n). \tag{2}$$

Описанный процесс нахождения значения выражения (2) будет продолжаться бесконечно в следующих случаях:

а) значение $f(x_1, \ldots, x_{n-1}, 0)$ не определено;

б) значения $f(x_1, \ldots, x_{n-1}, y)$ для $y = 0, 1, \ldots, a - 1$ определены, но отличны от x_n, а значение $f(x_1, \ldots, x_{n-1}, a)$ не определено;

в) значения $f(x_1, \ldots, x_{n-1}, y)$ определены для всех $y = 0, 1, 2, \ldots$ и отличны от x_n.

Во всех этих случаях значение выражения (2) считается неопределенным. В остальных случаях описанный процесс обрывается и дает наименьшее решение $y = a$ уравнения (1). Это решение, как сказано, и будет значением выражения (2).

Например, мы уже условились обозначать через $x \dot{-} y$ функцию, равную обычной разности $x - y$ для $x \geqslant y$ и нулю для $x < y$. Поэтому, в согласии с указанным

смыслом символа μ, для всех x, y имеем

$$x - y = \mu_z (y + z = x). \tag{3}$$

Аналогично,

$$\mu_x (\text{sg}\, x = 1) = 1,$$

$$\mu_y (y \dot{-} x = 0) = 0.$$

Это соответственно наименьшие решения уравнений $\text{sg}\, x = 1$ и $y \dot{-} x = 0$.

Напротив, значение выражения

$$\mu_y(y (y - (x + 1)) = 0) \tag{4}$$

неопределенное, так как уже значение терма $0 \cdot (0 - (x + 1))$ неопределенное. В то же время уравнение

$$y (y - (x + 1)) = 0$$

имеет решение $y = x + 1$, но оно не совпадает с значением выражения (4). Этот пример показывает, что для частичных функций $f (x_1, \ldots, x_{n-1}, y)$ выражение (2), строго говоря, не есть наименьшее решение уравнения (1). Если же функция $f (x_1, \ldots, x_{n-1}, y)$ всюду определенная и уравнение (1) имеет решения, то (2) есть наименьшее решение для (1).

Значение выражения (2) при заданной функции f зависит от выбора значений для параметров $x_1, \ldots, x_{n-1}, x_n$ и потому оно есть функция (частичная) от аргументов $x_1, \ldots, x_n$. Эту функцию мы будем обозначать символически через $\boldsymbol{M}f$, где $\boldsymbol{M}$ — символ операции, переводящей функцию f в функцию $\boldsymbol{M}f$. Если заданная функция f одноместная, то функцию $\boldsymbol{M}f$ обозначают часто через f^{-1} и называют *обращением* функции f или, короче, обратной функцией. Таким образом,

$$f^{-1} (x) = \mu_y (f (y) = x).$$

Например, для функций sg и $\boldsymbol{s}$ обратными будут функции

$$\text{sg}^{-1}\, x = \begin{cases} x, & \text{если } x = 0, 1, \\ \text{неопр.,} & \text{если } x > 1, \end{cases}$$

и соответственно

$$\boldsymbol{s}^{-1} (x) = \begin{cases} x - 1, & \text{если } x > 0, \\ \text{неопр.,} & \text{если } x = 0. \end{cases}$$

Для многоместных функций f запись f^{-1} не употребительна. В дальнейшем оператор $\boldsymbol{M}$ будет называться *оператором минимизации.* Например, из формулы (3) получается

$$\boldsymbol{M}(+) = \boldsymbol{S}^3(-, I_2^2, I_1^2).$$

Наряду с выражениями вида (2) далее нам будут встречаться выражения вида

$$\mu_y(f(x_1, \ldots, x_n, y) = g(x_1, \ldots, x_n, y)),$$
$$\mu_y(f(x_1, \ldots, x_n, y) \neq g(x_1, \ldots, x_n, y)).$$

Значения их считаются, по определению, совпадающими с значениями соответственно выражений

$$\mu_y(|f(x_1, \ldots, x_n, y) - g(x_1, \ldots, x_n, y)| = 0),$$
$$\mu_y(\mathrm{sg}\,|f(x_1, \ldots, x_n, y) - g(x_1, \ldots, x_n, y)| = 1).$$

Мы хотим теперь ввести следующее

Основное определение. *Частичная функция f называется частично рекурсивной относительно системы частичных функций $\mathfrak{S}$, если f может быть получена из функций системы $\mathfrak{S}$ и простейших функций $\boldsymbol{o}$, $\boldsymbol{s}$, I_m^n конечным числом операций подстановки, примитивной рекурсии и минимизации.*

Частичная функция f называется частично рекурсивной, если она может быть получена из простейших функций $\boldsymbol{o}$, $\boldsymbol{s}$, I_m^n конечным числом операций подстановки, примитивной рекурсии и минимизации.

На языке термов это определение может быть выражено следующим образом: частичная функция f называется частично рекурсивной относительно системы частичных функций $\mathfrak{S}$, если f есть значение (операторного) терма, записанного с помощью операторных символов $\boldsymbol{R}$, $\boldsymbol{M}$, $\boldsymbol{S}^i$, предметных символов $\boldsymbol{o}$, $\boldsymbol{s}$, I_m^n простейших функций и предметных символов для функций из системы $\mathfrak{S}$.

Частичная функция f называется частично рекурсивной, если она является значением (операторного) терма, записанного с помощью операторных символов $\boldsymbol{R}$, $\boldsymbol{M}$, $\boldsymbol{S}^i$ и символов $\boldsymbol{o}$, $\boldsymbol{s}$, I_m^n простейших функций.

Из указанного основного определения непосредственно вытекают следующие свойства относительно частично рекурсивных функций.

1) Каждая частичная функция, примитивно рекурсивная относительно системы функций $\mathfrak{S}$, является и частично рекурсивной относительно $\mathfrak{S}$. В частности, все примитивно рекурсивные функции частично рекурсивны.

2) Класс частично рекурсивных функций шире класса примитивно рекурсивных функций, так как все примитивно рекурсивные функции всюду определенные, а среди частично рекурсивных функций встречаются и функции не всюду определенные, например функции sg^{-1}, s^{-1}, а также нигде не определенная функция

$$f(x) = \mu_z(x + 1 + z = 0).$$

3) Операции подстановки, примитивной рекурсии и минимизации, произведенные над функциями, частично рекурсивными относительно системы $\mathfrak{S}$, дают в результате функции, снова частично рекурсивные относительно $\mathfrak{S}$. В частности, каждый (функциональный) терм, записанный с помощью символов функций, частично рекурсивных относительно $\mathfrak{S}$, и предметных переменных $x_1, \ldots, x_n$, представляет функцию от $x_1, \ldots, x_n$, частично рекурсивную относительно $\mathfrak{S}$.

Уже из примеров, приведенных в п. 2.2, видно, что многие числовые функции примитивно рекурсивны. Чтобы получить первое, грубое представление о связи между примитивно рекурсивными и частично рекурсивными функциями, введем следующие понятия.

Характеристической функцией χ_A какого-нибудь множества натуральных чисел A называется одноместная функция, равная 0 в точках множества A и равная 1 в точках, не принадлежащих A. *Частичной характеристической функцией* множества A называется функция, равная 0 в точках множества A и не определенная в точках, не принадлежащих A.

Например, характеристическая функция пустого множества есть функция, равная 1 для любого значения аргумента, а частичная характеристическая функция пустого множества есть нигде не определенная функция. Вообще легко понять, что характеристическая и частичная характеристическая функции совпадают лишь для множества всех натуральных чисел.

Множество натуральных чисел A называется *примитивно рекурсивным*, если его характеристическая функция примитивно рекурсивна. Множество A называется *частично рекурсивным*, если его частичная характеристическая функция частично рекурсивна. Аналогично определяются понятия *примитивно рекурсивного* и *частично рекурсивного* множеств *относительно системы функций* $\mathfrak{S}$.

Покажем, что *каждое (относительно) примитивно рекурсивное множество является (относительно) частично рекурсивным*. Действительно, пусть $\chi(x)$ — характеристическая функция какого-нибудь множества натуральных чисел A. Тогда функция $\chi_{ч}(x)$, определяемая равенством

$$\chi_{ч}(x) = 0 - \chi(x), \tag{5}$$

будет частичной характеристической функцией множества A. Так как операция вычитания частично рекурсивна, то из формулы (5) следует, что функция $\chi_{ч}(x)$ также частично рекурсивна.

Детальное изучение свойств частично рекурсивных функций составляет центральную задачу теории рекурсивных функций. Оно будет проведено в главе III. В качестве предварительной информации о свойствах частично рекурсивных функций может служить следующая очевидная

Теорема 1. *Пусть $f(x)$ — какая-нибудь примитивно рекурсивная функция и A — произвольное примитивно рекурсивное множество натуральных чисел. Тогда частичная функция $f_{ч}(x)$, определенная схемой*

$$f_{ч}(x) = \begin{cases} f(x), & \text{если } x \in A, \\ \text{неопр.}, & \text{если } x \notin A, \end{cases} \tag{6}$$

является частично рекурсивной.

В самом деле, по доказанному выше частичная характеристическая функция $\chi_{ч}(x)$ множества A частично рекурсивна. Из схемы (6) вытекает, что для всех значений x

$$f_{ч}(x) = f(x) + \chi_{ч}(x)$$

и, значит, функция $f_{ч}(x)$ частично рекурсивная.

Теорема 1 дает возможность строить многочисленные примеры частично рекурсивных функций.

Понятие частично рекурсивной функции — одно из главных понятий теории алгоритмов. Значение его уже было указано во введении. Кратко говоря, это значение состоит в следующем. С одной стороны, каждая стандартно заданная, например, посредством указанного выше операторного терма, частично рекурсивная функция вычислима путем определенной процедуры механического характера, которая несомненно отвечает нашему интуитивному представлению об алгоритмах. С другой стороны, какие бы классы точно очерченных «алгоритмов» до сих пор фактически ни строились, во всех случаях неизменно оказывалось, что числовые функции, вычислимые посредством алгоритмов этих классов, были частично рекурсивными. Поэтому ныне общепринятой является следующая естественнонаучная гипотеза, обычно формулируемая как

Тезис Чёрча. *Класс алгоритмически (или машинно) вычислимых частичных числовых функций совпадает с классом всех частично рекурсивных функций.*

Этот тезис дает алгоритмическое истолкование понятию частично рекурсивной функции. Несколько сложнее дело обстоит с истолкованием понятия частичной рекурсивности относительно заданной системы функций $\mathfrak{S}$. Такое истолкование было впервые указано Тьюрингом. Для простоты предположим, что система $\mathfrak{S}$ состоит лишь из одной функции $h(x)$. Если значения этой функции вычислимы посредством некоторого алгоритма и, значит, ввиду тезиса Чёрча, функция $h(x)$ частично рекурсивная, то каждая функция f, частично рекурсивная относительно h, будет просто частично рекурсивной. Если заданная функция h не частично рекурсивная, то единого алгоритма для вычисления произвольного значения функции h нет, и вычислить какое-нибудь значение функции $h(x)$ — это математическая проблема. Пусть теперь нам задана какая-нибудь функция f, частично рекурсивная относительно h. Это значит, что функция f может быть представлена в виде значения операторного терма, содержащего символ функции h и символы простейших функций. Мы предположим, что этот терм $\mathfrak{a}$ нам известен. В таком случае, просматривая определения операций $\boldsymbol{R}$, $\boldsymbol{M}$, $\boldsymbol{S}^i$, мы легко убеждаемся, что вместе с термом $\mathfrak{a}$ определен и алгоритм, который позволяет вычислять значения функции f *при условии, что мы можем*

находить те значения функции h (путем решения соответствующих проблем), *которые указывает алгоритм.*

Говорят, что функция f *алгоритмически вычислима относительно функции* h, если существует алгоритм, позволяющий вычислять значения функции f при условии, что мы можем находить те значения функции h, которые указываются алгоритмом. При этом значения функции h, необходимые на более поздних шагах алгоритма, могут зависеть от значений h, требовавшихся на предыдущих шагах алгоритма.

Это определение относительной алгоритмической вычислимости, конечно, не полное. Смысл его зависит от того, какой смысл мы вкладываем в понятие обычного (не относительного) алгоритма, и других обстоятельств. Как и понятие алгоритма, понятие относительного алгоритма можно уточнять различными способами. Однако при всех фактически испробованных уточнениях оказывалось, что относительно вычислимые функции являлись относительно частично рекурсивными. Поэтому по аналогии с тезисом Чёрча обычно принимается в качестве естественнонаучной гипотезы и

Тезис Тьюринга. *Класс функций, алгоритмически вычислимых относительно какой-либо функции* h *(или класса функций* $\mathfrak{S}$*), совпадает с классом частичных функций, частично рекурсивных относительно* h *(соответственно, относительно системы* $\mathfrak{S}$*).*

Из тезиса Тьюринга вытекает тезис Чёрча. Обратное, по-видимому, нельзя считать справедливым. Конечно, здесь речь идет не о строгой логической зависимости, а скорее о зависимости в некотором философском смысле.

2.4. Общерекурсивные функции. Как уже говорилось, для каждой частично рекурсивной функции f существует механический процесс, посредством которого любое натуральное число x перерабатывается в значение $f(x)$ функции f. Этот процесс продолжается бесконечно, не давая окончательного результата, тогда и только тогда, когда значение функции f в точке x не определено. Таким образом, всюду определенные частично рекурсивные функции — это функции, для вычисления значений которых существует алгоритм, обрывающийся через конечное число шагов для любого начального числа. Алгоритмы, которые

перерабатывают в определенное число любое заданное число, играют особую роль в теории алгоритмов. Вместе с ними особое положение в теории рекурсивных функций занимают и всюду определенные частично рекурсивные функции.

Следующий способ получения всюду определенных частично рекурсивных функций вполне очевиден. Операция минимизации, введенная в п. 2.3, ставит в соответствие любой заданной функции f определенную частичную функцию $\boldsymbol{M}f$. Введем теперь еще одну операцию, которую временно будем обозначать символом $\boldsymbol{M}^l$ и называть слабой минимизацией. По определению мы положим

$$\boldsymbol{M}^l f = \boldsymbol{M}f,$$

если функция $\boldsymbol{M}f$ всюду определенная. Если же эта функция определена не всюду, то значение $\boldsymbol{M}^l f$ будем считать неопределенным.

Функции, которые можно получить из простейших функций $\boldsymbol{o}$, $\boldsymbol{s}$, I^n_m конечным числом операций подстановки, примитивной рекурсии и слабой минимизации, называются *общерекурсивными*.

Так как операции $\boldsymbol{R}$, $\boldsymbol{M}^l$, $\boldsymbol{S}^i$, выполненные над всюду определенными функциями, либо ничего не дают, либо дают снова функции, всюду определенные, то все *общерекурсивные функции всюду определенные*.

С другой стороны, если результат операции слабой минимизации определен, то он совпадает с результатом операции обычной минимизации. Поэтому все *общерекурсивные функции являются всюду определенными частично рекурсивными функциями*.

Обратное тоже верно: каждая всюду определенная частично рекурсивная функция общерекурсивна. Однако этот результат является следствием довольно тонкой теоремы, которая будет доказана лишь в п. 6.1. Заметим пока, что согласно определению каждая примитивно рекурсивная функция является общерекурсивной.

В п. 5.2 будут построены общерекурсивные функции, не являющиеся примитивно рекурсивными. Таким образом, класс общерекурсивных функций шире класса примитивно рекурсивных функций.

Примеры и задачи

1. Операция подстановки одноместной функции в одноместную функцию дает снова одноместную функцию. Эту операцию мы будем обозначать символом $*$. Таким образом, по определению,

$$f * g = S^2(f, g),$$
$$(f * g)(x) = f(g(x)).$$

Операция $*$ ассоциативна, но не коммутативна:

$$f * (g * h) = (f * g) * h,$$
$$\boldsymbol{s} * \mathrm{sg} \neq \mathrm{sg} * \boldsymbol{s}.$$

Показать, что для любой одноместной функции f

$$I_1^1 * f = f * I_1^1 = f * f^{-1} * f = f, \qquad f^{-1} * f * f^{-1} = f^{-1}.$$

Если f^{-1} всюду определенная, то $f * f^{-1} = I_1^1$. Если выражение $f^{-1}(x)$ для некоторого x определено, то

$$(f * f^{-1})(x) = x.$$

Показать, что существуют одноместные функции f, для которых f^{-1} всюду определена и в то же время $f^{-1} * f \neq I_1^1$.

2. Для двуместных функций вводим операцию τ, полагая

$$f^{\tau}(x, y) = f(y, x).$$

Операция τ удовлетворяет следующим соотношениям:

$$f^{\tau} = S^3(f, I_2^2, I_1^2),$$
$$\mu_x(f(x, y) = z) = (\boldsymbol{M} f^{\tau})(y, z).$$

3. Если x — вещественное число, то символом $[x]$ будем обозначать целую часть x, то есть наибольшее целое число, не превосходящее x. Показать, что функция

$$\boldsymbol{q}(x) = x - [\sqrt{x}]^2$$

удовлетворяет соотношениям

$$\boldsymbol{q}^{-1}(2x) = x^2 + 2x,$$
$$\boldsymbol{q}^{-1}(2x + 1) = x^2 + 4x + 2.$$

4. Пусть $\mathfrak{F}$ — множество всех частичных функций от любого числа переменных. Рассмотрим частичную алгебру

$$\mathfrak{A} = \langle \mathfrak{F};\ \boldsymbol{M},\ \boldsymbol{R},\ \boldsymbol{S}^2,\ \boldsymbol{S}^3,\ \ldots \rangle.$$

Основное определение частичной рекурсивности дает: совокупность всех частичных функций, частично рекурсивных относительно заданной системы частичных функций $\mathfrak{S}$, есть подалгебра алгебры $\mathfrak{A}$, порожденная в $\mathfrak{A}$ системой $\mathfrak{S}$, I_m^n, $\boldsymbol{o}$, $\boldsymbol{s}$.

Обозначим через $\mathfrak{F}_l$ совокупность всех всюду определенных функций из $\mathfrak{F}$ и рассмотрим частичную подалгебру

$$\mathfrak{A}_l = \langle \mathfrak{F}_l;\ \boldsymbol{M},\ \boldsymbol{R},\ \boldsymbol{S}^2,\ \boldsymbol{S}^3,\ \ldots \rangle$$

алгебры $\mathfrak{A}$. Совокупность всех общерекурсивных функций есть подалгебра алгебры $\mathfrak{A}_l$, порожденная в $\mathfrak{A}_l$ простейшими функциями $\boldsymbol{o}$, $\boldsymbol{I}^n_m$, $\boldsymbol{s}$.

5. Доказать, что каждая всюду определенная функция, равная натуральному числу a везде, за исключением конечного числа точек, является примитивно рекурсивной функций.

6. Доказать примитивную рекурсивность двуместных функций $[x/y]$, rest (x, y), где $[x/y]$ обозначает (неполное) частное, а rest (x, y) — остаток от деления x на y. По определению полагаем также, что $[x/0] = x$ и rest $(x, 0) = x$.

7. Если значения примитивно рекурсивной, общерекурсивной или частично рекурсивной функции изменить лишь на конечном множестве точек, то новая функция будет снова примитивно рекурсивной, общерекурсивной или соответственно частично рекурсивной.

8. Показать, что примитивно рекурсивна каждая

а) конечная совокупность чисел;

б) совокупность чисел вида $an + b$ $(n = 0, 1, 2, \ldots)$;

в) совокупность чисел вида $a \cdot b^n$ $(n = 0, 1, 2, \ldots)$.

9. Показать, что область определенности одноместной частично рекурсивной функции есть множество частично рекурсивное. Совокупность значений n-местной частично рекурсивной функции является частично рекурсивным множеством.

ГЛАВА II

ПРИМИТИВНО РЕКУРСИВНЫЕ ФУНКЦИИ И РЕКУРСИВНО ПЕРЕЧИСЛИМЫЕ МНОЖЕСТВА

В первой половине главы продолжается изучение свойств примитивно рекурсивных функций, точное определение которых уже было дано в предшествующей главе. Во второй половине главы вводится новое фундаментальное понятие — понятие рекурсивно перечислимого множества натуральных чисел и доказывается ряд свойств этих множеств, составляющих основу, на которой строится дальнейшая теория рекурсивных функций.

§ 3. ПРИМИТИВНО РЕКУРСИВНЫЕ ФУНКЦИИ

В п.2.2 введено понятие примитивно рекурсивной функции и начато изучение свойств этого понятия. Теперь изучение свойств примитивно рекурсивных функций будет продолжено, причем главной целью будет доказательство примитивной рекурсивности ряда наиболее часто встречающихся числовых функций.

3.1. Операции суммирования и мажорированного обращения. Согласно п. 2.2 операции подстановки и примитивной рекурсии, произведенные над примитивно рекурсивными функциями, дают в качестве результата снова примитивно рекурсивные функции. Мы теперь определим еще несколько операций, обладающих этим свойством.

Теорема 1. *Пусть n-местная (частичная) функция g примитивно рекурсивна (относительно какой-нибудь системы частичных функций $\mathfrak{S}$). Тогда n-местная функция f,*

определенная равенством

$$f(x_1, \ldots, x_n) = \sum_{i=0}^{x_n} g(x_1, \ldots, x_{n-1}, i),$$

также примитивно рекурсивна (относительно $\mathfrak{S}$).

Действительно, из указанного равенства вытекает

$$f(x_1, \ldots, x_{n-1}, 0) = g(x_1, \ldots, x_{n-1}, 0),$$
$$f(x_1, \ldots, x_{n-1}, y+1) = \\ = f(x_1, \ldots, x_{n-1}, y) + g(x_1, \ldots, x_{n-1}, y+1).$$

Следовательно, функция f возникает посредством примитивной рекурсии из примитивно рекурсивных функций

$$g(x_1, \ldots, x_{n-1}, 0), \; h(x_1, \ldots, x_{n-1}, y, z) = \\ = z + g(x_1, \ldots, x_{n-1}, y+1)$$

и потому f примитивно рекурсивная.

Следствие 1. *Если n-местная функция g примитивно рекурсивна (относительно* $\mathfrak{S}$), *то* $(n+1)$*-местная функция f, определенная схемой*

$$f(x_1, \ldots, x_{n-1}, y, z) = \begin{cases} \sum\limits_{i=y}^{z} g(x_1, \ldots, x_{n-1}, i), & \text{если } y \leqslant z; \\ 0 & , \text{если } y > z, \end{cases} \quad (1)$$

также примитивно рекурсивна (относительно $\mathfrak{S}$).

Из схемы (1) видно, что

$$f(x_1, \ldots, x_{n-1}, y, z) = \\ = \left(\sum_{i=0}^{z} g(x_1, \ldots, x_{n-1}, i) \dot{-} \sum_{i=0}^{y} g(x_1, \ldots, x_{n-1}, i)\right) + \\ + g(x_1, \ldots, x_{n-1}, y) \cdot \overline{\mathrm{sg}}\,(y \dot{-} z).$$

Так как операция $\dot{-}$ примитивно рекурсивна, то в силу теоремы 1 функция f также примитивно рекурсивна.

Иногда говорят, что определенная в теореме 1 функция f возникает из функции g операцией суммирования. Теорема 1 означает, что операция суммирования, произведенная над примитивно рекурсивной функцией, дает снова примитивно рекурсивную функцию.

Следствие 2. *Если g, h, k — примитивно рекурсивные функции, то функция f^*, определенная соотношением*

$$f^*(x_1, \ldots, x_n) = \sum_{i=h(x_1, \ldots, x_n)}^{k(x_1, \ldots, x_n)} g(x_1, \ldots, x_{n-1}, i), \qquad (2)$$

также примитивно рекурсивна.

Формула (2) равносильна формуле

$$f^*(x_1, \ldots, x_n) = f(x_1, \ldots, x_{n-1}, h, k),$$

где f — функция, определенная схемой (1). Так как f, h, k — примитивно рекурсивные, то f^* также примитивно рекурсивна.

Аналогично теореме 1 доказывается и

Теорема 2. *Если n-местная функция g примитивно рекурсивна (относительно системы функций $\mathfrak{S}$), то n-местная функция, определенная формулой*

$$f(x_1, \ldots, x_{n-1}, x_n) = \prod_{i=0}^{x_n} g(x_1, \ldots, x_{n-1}, i),$$

также примитивно рекурсивна (относительно $\mathfrak{S}$).

В этом случае говорят, что функция $f(x_1, \ldots, x_n)$ получается из функции $g(x_1, \ldots, x_n)$ операцией мультиплицирования.

Доказательство ввиду его простоты мы опустим.

Далее часто будет встречаться так называемое кусочное задание функции. Заключается оно в следующем. Пусть заданы некоторые функции $f_i(x_1, \ldots, x_n)$, $i = 1, \ldots, s+1$, и указаны какие-то условия $P_j(x_1, \ldots, x_n)$ ($j = 1, \ldots, s$), которые для любой системы чисел $x_1, \ldots, x_n$ могут быть истинны или ложны. Допустим, далее, что ни для одной системы чисел $x_1, \ldots, x_n$ никакие два из упомянутых условий не могут быть одновременно истинными. Функция $f(x_1, \ldots, x_n)$, заданная схемой

$$f(x_1, \ldots, x_n) = \begin{cases} f_1(x_1, \ldots, x_n), \text{ если } P_1(x_1, \ldots, x_n) \text{ истинно;} \\ f_2(x_1, \ldots, x_n), \text{ если } P_2(x_1, \ldots, x_n) \text{ истинно;} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ f_s(x_1, \ldots, x_n), \text{ если } P_s(x_1, \ldots, x_n) \text{ истинно;} \\ f_{s+1}(x_1, \ldots, x_n) \text{ для остальных } x_1, \ldots, x_n, \end{cases}$$

называется *функцией, определенной кусочной схемой*. Будет ли функция f примитивно рекурсивной, зависит от природы функций f_i и условий P_j. Простейший случай, когда функция f будет заведомо примитивно рекурсивной, дает

Теорема 3. *Пусть заданы n-местные примитивно рекурсивные (относительно $\mathfrak{S}$) функции $f_1, \ldots, f_{s+1}, \alpha_1, \ldots, \alpha_s$, причем ни при каких значениях переменных никакие две из функций $\alpha_1, \ldots, \alpha_s$ одновременно в 0 не обращаются. Тогда функция f, определенная кусочной схемой*

$$f(x_1, \ldots, x_n) = \begin{cases} f_1(x_1, \ldots, x_n), \text{ если } \alpha_1(x_1, \ldots, x_n) = 0, \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ f_s(x_1, \ldots, x_n), \text{ если } \alpha_s(x_1, \ldots, x_n) = 0, \\ f_{s+1}(x_1, \ldots, x_n) \text{ в остальных случаях,} \end{cases}$$

будет примитивно рекурсивной (относительно $\mathfrak{S}$).

Для доказательства достаточно заметить, что функцию f можно представить в виде

$$f = f_1 \overline{\mathrm{sg}}\, \alpha_1 + \ldots + f_s \overline{\mathrm{sg}}\, \alpha_s + f_{s+1} \mathrm{sg}\, (\alpha_1 \ldots \alpha_s),$$

где sg, $\overline{\mathrm{sg}}$ — введенные в п. 2.2 примитивно рекурсивные одноместные функции.

В теореме 3 рассмотрен типичный случай, когда условия P_i имеют вид $\alpha_i = 0$. Так как условия вида

$$\alpha_i = \beta_i, \ \alpha_i \leqslant \beta_i, \ \alpha_i < \beta_i \tag{3}$$

равносильны, соответственно, условиям

$$|\alpha_i - \beta_i| = 0, \ \alpha_i \dot{-} \beta_i = 0, \ \overline{\mathrm{sg}}\, (\beta_i \dot{-} \alpha_i) = 0,$$

то теорема 3 останется справедливой и в том случае, когда в кусочной схеме равенства $\alpha_i = 0$ заменяются по произволу требованиями вида (3), где α_i, β_i — примитивно рекурсивные функции.

Рассмотрим уравнение

$$g(x_1, \ldots, x_n, y) = 0, \tag{4}$$

левая часть которого есть всюду определенная функция. Допустим, что для каждых значений $x_1, \ldots, x_n$ уравнение (4) имеет единственное решение y. Тогда это решение будет однозначной всюду определенной функцией от

$x_1, \ldots, x_n$. Спрашивается, будет ли эта функция $y = f(x_1, \ldots, x_n)$ примитивно рекурсивной, если левая часть уравнения (4) есть примитивно рекурсивная функция от $x_1, \ldots, x_n, y$? В главе IV будет показано, что в общем случае ответ на этот вопрос отрицателен. Это не исключает, конечно, возможности, что в отдельных случаях ответ будет положительным.

Теорема 4 (о мажорируемых неявных функциях). *Пусть* $g(x_1, \ldots, x_n, y)$, $\alpha(x_1, \ldots, x_n)$ — *такие примитивно рекурсивные функции, что уравнение*

$$g(x_1, \ldots, x_n, y) = 0 \tag{5}$$

для каждых $x_1, \ldots, x_n$ *имеет по меньшей мере одно решение и*

$$\mu_y(g(x_1, \ldots, x_n, y) = 0) \leqslant \alpha(x_1, \ldots, x_n) \tag{6}$$

для любых $x_1, \ldots, x_n$. *Тогда функция*

$$f(x_1, \ldots, x_n) = \mu_y(g(x_1, \ldots, x_n, y) = 0)$$

также примитивно рекурсивна.

Фиксируем какие-нибудь значения $x_1, \ldots, x_n$ и пусть

$$a = \mu_y(g(x_1, \ldots, x_n, y) = 0).$$

Рассмотрим последовательность произведений

$$g(x_1, \ldots, x_n, 0)\, g(x_1, \ldots, x_n, 1) \ldots g(x_1, \ldots, x_n, i) \quad (i = 0, 1, 2, \ldots). \tag{7}$$

Так как $y = a$ есть наименьшее решение уравнения (5), то первые a членов последовательности (7) нулю не равны, а все остальные члены содержат равный нулю сомножитель $g(x_1, \ldots, x_n, a)$ и потому равны нулю. Поэтому из условия (6) вытекает, что

$$a = \sum_{i=0}^{\alpha(x_1, \ldots, x_n)} \operatorname{sg}(g(x_1, \ldots, x_n, 0) \cdot \ldots \cdot g(x_1, \ldots, x_n, i)). \tag{8}$$

Вводя примитивно рекурсивную функцию

$$h(x_1, \ldots, x_n, z) = \prod_{i=0}^{z} g(x_1, \ldots, x_n, i),$$

мы сможем переписать равенство (8) в форме

$$f(x_1, \ldots, x_n) = \sum_{i=0}^{\alpha(x_1, \ldots, x_n)} \operatorname{sg} h(x_1, \ldots, x_n, i).$$

В силу теорем 1, 2 отсюда следует, что функция f примитивно рекурсивна.

3.2. Примитивная рекурсивность некоторых арифметических функций. При помощи изложенных в предыдущем пункте общих теорем теперь можно легко проверить, что обычные арифметические функции, связанные с отношениями делимости и простоты натуральных чисел, являются примитивно рекурсивными.

Начнем с частного и остатка от деления числа x на число y, которые будем обозначать через $[x/y]$ и $\operatorname{rest}(x, y)$. Чтобы иметь дело с всюду определенными функциями, дополнительно положим, что

$$[x/0] = x, \ \operatorname{rest}(x, 0) = x$$

для всех x. Ясно, что так определенные функции связаны тождеством

$$\operatorname{rest}(x, y) = x \dot{-} (y \cdot [x/y])$$

и, значит, из примитивной рекурсивности функции $[x/y]$ будет вытекать и примитивная рекурсивность функции $\operatorname{rest}(x, y)$.

Согласно определению, при $y > 0$ число $[x/y] = n$ удовлетворяет соотношениям

$$ny \leqslant x < (n+1)y.$$

Отсюда видно, что n равно числу нулей в последовательности

$$1y \dot{-} x, \ 2y \dot{-} x, \ldots, ny \dot{-} x, \ldots, xy \dot{-} x.$$

Поэтому для $y > 0$ имеем формулу

$$[x/y] = \sum_{i=1}^{x} \overline{\operatorname{sg}}(iy \dot{-} x). \qquad (1)$$

Непосредственная проверка показывает, что формула (1) верна и при $y = 0$. Так как функция $\overline{\operatorname{sg}}(iy \dot{-} x)$, стоящая под знаком суммы, примитивно рекурсивная, то на

основании теоремы 1 заключаем, что функция $[x/y]$, а вместе с ней и функция rest (x, y) примитивно рекурсивны.

Говорят, что число x делится (без остатка) на число y, или что *y делит x*, если rest $(x, y) = 0$. Введем двуместную функцию div (x, y), полагая по определению

$$\operatorname{div}(x, y) = \begin{cases} 1, & \text{если } \operatorname{rest}(x, y) = 0, \\ 0, & \text{если } \operatorname{rest}(x, y) \neq 0. \end{cases}$$

Очевидное соотношение

$$\operatorname{div}(x, y) = \overline{\operatorname{sg}} \operatorname{rest}(x, y)$$

показывает, что функция div примитивно рекурсивна.

Полагаем также по определению

$$\operatorname{nd} x = \sum_{i=0}^{x} \operatorname{div}(x, i). \qquad (2)$$

Если $x \neq 0$, то делители числа x не превосходят x и потому для положительных значений x число nd x совпадает с числом различных делителей x (включая число 1). Из формулы (2) видно, что функция nd примитивно рекурсивна.

В арифметике натуральное число x называется *простым*, если оно имеет точно 2 различных делителя. Число 0 имеет бесконечно много, а число 1 лишь один делитель. Поэтому 0 и 1 — не простые числа. Для простых чисел 2, 3, 5, 7, ... введем стандартные обозначения $p_0 = 2$, $p_1 = 3$, ... Таким образом, p_n есть $(n + 1)$-е простое число в натуральном ряде чисел.

Обозначим через $\chi_p(n)$ характеристическую функцию свойства быть простым числом. Иначе говоря, положим $\chi_p(n) = 0$ для n простого и $\chi_p(n) = 1$ для n непростого. Так как простые числа и только они имеют ровно 2 делителя, то

$$\chi_p(x) = \operatorname{sg} |\operatorname{nd} x - 2|$$

и, следовательно, функция $\chi_p(x)$ примитивно рекурсивна.

Одной из наиболее известных арифметических функций является функция $\pi(x)$, равная числу простых чисел, не превосходящих x. Очевидная формула

$$\pi(x) = \sum_{i=0}^{x} \overline{\operatorname{sg}} \chi_p(x)$$

показывает, что функция $\pi(x)$ примитивно рекурсивна.

Из определения функции $\pi(x)$ непосредственно следует, что

$$\pi(p_n) = n + 1,$$
$$\pi(x) < n + 1, \quad \text{если } x < p_n.$$

Отсюда видно, что $x = p_n$ есть минимальное решение уравнения $\pi(x) = n + 1$. Поэтому

$$p_n = \boldsymbol{p}(n) = \mu_x(|\pi(x) - (n+1)| = 0).$$

Стоящая под знаком μ-операции функция $|\pi(x) - (n+1)|$ примитивно рекурсивна. В силу теоремы о мажорируемых неявных функциях (п. 3.1) для доказательства примитивной рекурсивности функции $\boldsymbol{p}(n)$ остается лишь найти примитивно рекурсивную функцию $\alpha(n)$ такую, чтобы для всех n было $p_n \leqslant \alpha(n)$. Из теории чисел хорошо известно, что в качестве функции $\alpha(n)$ можно взять 2^{2^n}.

В самом деле, требующееся нам неравенство

$$p_n \leqslant 2^{2^n} \tag{3}$$

заведомо истинно при $n = 0$. По индукции далее предполагаем, что неравенство (3) истинно для всех значений n, меньших некоторого числа $s + 1$. Докажем, что (3) истинно и для $n = s + 1$.

По предположению имеем

$$p_0 p_1 \ldots p_s + 1 \leqslant 2^{2^0+2^1+\ldots+2^s} + 1 < 2^{2^{s+1}}.$$

Число $a = p_0 p_1 \ldots p_s + 1$ больше единицы и потому должно иметь какой-то простой делитель p_r. Этот делитель не может совпадать ни с одним из простых чисел $p_0, p_1, \ldots, p_s$, так как при делении числа a на любое из чисел $p_0, p_1, \ldots, p_s$ получается остаток 1. Но все простые числа, не превосходящие p_s, содержатся в последовательности $p_0, p_1, \ldots, p_s$. Число p_r в нее не входит и потому $p_{s+1} \leqslant p_r$. Так как $p_r \leqslant a$, то

$$p_{s+1} \leqslant a \leqslant 2^{2^{s+1}},$$

что и требовалось.

Итак, неравенство (3) доказано, а вместе с ним доказана и примитивная рекурсивность функции $\boldsymbol{p}(x) = p_x$.

Введем еще одну важную для дальнейшего функцию. Эту функцию мы будем обозначать через ex (x, y) или, сокращенно, $\mathrm{ex}_x y$ и называть *экспонентой* числа p_x в числе y. По определению, при $y \neq 0$ $\mathrm{ex}_x y$ полагаем равным показателю наивысшей степени простого числа p_x, на которую делится y. Для $y = 0$ полагаем по определению $\mathrm{ex}_x\, 0 = 0$ для всех значений x. Например,

$$\mathrm{ex}_0\, 8 = 3, \quad \mathrm{ex}_1\, 8 = 0, \; \mathrm{ex}_0\, 0 = 0.$$

Каждое натуральное число $n > 1$ можно однозначно представить в виде произведения

$$n = p_{i_0}^{a_0} p_{i_1}^{a_1} \ldots p_{i_s}^{a_s} \qquad (a_j > 0, \; i_0 < i_1 < \ldots < i_s)$$

положительных степеней различных простых чисел. Отсюда в силу определения функции ex получаем

$$\mathrm{ex}_{i_0} n = a_0, \; \ldots, \; \mathrm{ex}_{i_s} n = a_s$$

и $\mathrm{ex}_i\, n = 0$ для всех значений i, отличных от $i_0, \ldots, i_s$.

Чтобы доказать примитивную рекурсивность функции ex (x, y), мы снова воспользуемся теоремой о мажорируемой неявной функции. По определению, ex $(x, y + 1)$ есть наибольшее значение u, для которого p_x^u есть делитель $y + 1$. Поэтому можно утверждать также, что ex $(x, y + 1)$ есть наименьшее значение u, для которого p_x^{u+1} не делит $y + 1$, то есть

$$\mathrm{ex}\,(x, y+1) = \mu_u\, (\overline{\mathrm{sg}}\, \mathrm{rest}\, (y+1, p_x^{u+1}) = 0). \tag{4}$$

Функция p_x^{u+1} примитивно рекурсивна и, кроме того,

$$\mathrm{ex}\,(x, y+1) \leqslant y+1. \tag{5}$$

В силу упомянутой теоремы о мажорируемых неявных функциях, из (4) и (5) следует, что функция ex $(x, y + 1)$, а с нею и ex (x, y) = ex $(x, (y \dot{-} 1) + 1)$, примитивно рекурсивны.

Наконец, рассмотрим еще функцию

$$q(x) = x \dot{-} \left[\sqrt{x}\right]^2,$$

где символом $[z]$ обозначается целая часть вещественного числа z, равная наибольшему целому, не превосходящему z. Число $q\,(x)$ называется *квадратичным остатком* числа x.

Оно равно расстоянию от x до ближайшего слева точного квадрата.

Равенство $n = [\sqrt{x}]$ равносильно соотношению

$$n^2 \leqslant x < (n+1)^2,$$

где n — натуральное. Таким образом,

$$[\sqrt{x}] = \mu_t (\mathrm{sg}((t+1)^2 \dot{-} x) = 1)$$

и вместе с тем $[\sqrt{x}] \leqslant x$. По теореме о мажорируемой неявной функции отсюда следует, что функция $[\sqrt{x}]$ примитивно рекурсивная. Вместе с нею примитивно рекурсивной является и функция $q(x) = x \dot{-} [\sqrt{x}]^2$. Аналогичным образом доказывается примитивная рекурсивность функции $[\sqrt[n]{x}]$ и многих других арифметических функций.

Рекурсией называется такой способ задания функции, при котором значения определяемой функции для произвольных значений аргументов выражаются известным образом через значения определяемой функции для меньших значений аргументов. *Примитивная рекурсия* является одним из простейших видов этих более общих рекурсий. Более сложные рекурсии 2-й ступени будут рассмотрены в следующей главе, а здесь мы хотим в качестве примера на приложение введенных выше арифметических функций рассмотреть так называемую возвратную рекурсию, легко выражаемую через обычную примитивную рекурсию.

Определение. *Пусть* $\alpha_1(x), \ldots, \alpha_s(x)$ — *всюду определенные функции, удовлетворяющие для всех значений* x *условиям*

$$\alpha_i(x+1) \leqslant x \qquad (i = 1, \ldots, s). \tag{6}$$

Говорят, что функция $f(x_1, \ldots, x_{n+1})$ *получается возвратной рекурсией из функций* $g(x_1, \ldots, x_n)$, $h(x_1, \ldots, x_n, y, z_1, \ldots, z_s)$ *и вспомогательных функций* $\alpha_1, \ldots, \alpha_s$, *если для всех значений переменных* $x_1, \ldots, x_n, y$

$$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n),$$
$$f(x_1, \ldots, x_n, y+1) = h(x_1, \ldots, x_n, y, \tag{7}$$
$$f(x_1, \ldots, x_n, \alpha_1(y+1)), \ldots, f(x_1, \ldots, x_n, \alpha_s(y+1))).$$

Как и в случае примитивной рекурсии, легко доказывается, что для любых функций g, h, α_i, удовлетворяющих

высказанным в определении условиям, функция f, удовлетворяющая требованиям (7), существует и единственна. Более того, верна следующая

Теорема 1 (о возвратной рекурсии). *Функция f, возникающая из функций $g(x_1, \ldots, x_n)$, $h(x_1, \ldots, x_n, y, z_1, \ldots, z_s)$ и вспомогательных функций $\alpha_1(x), \ldots \ldots, \alpha_s(x)$, удовлетворяющих условиям* (6), *с помощью возвратной рекурсии, может быть получена из тех же функций g, h, α_i и простейших функций $\mathbf{o}$, $\mathbf{s}$, I_m^n операциями подстановки и примитивной рекурсии.*

Иными словами, функция f является примитивно рекурсивной относительно функций g, h, $\alpha_1, \ldots, \alpha_s$.

Для доказательства введем новую функцию F, полагая

$$F(x_1, \ldots, x_n, y) = \prod_{i=0}^{y} p_i^{f(x_1, \ldots, x_n, i)}. \tag{8}$$

Вспоминая определение функции $\mathrm{ex}(u, v)$, непосредственно заключаем, что для $u \leqslant y$

$$f(x_1, \ldots, x_n, u) = \mathrm{ex}(u, F(x_1, \ldots, x_n, y)). \tag{9}$$

По условию $\alpha_i(y+1) \leqslant y$. Поэтому соотношение

$$f(x_1, \ldots, x_n, \alpha_i(y+1)) = \mathrm{ex}(\alpha_i(y+1), F(x_1, \ldots, x_n, y)) \tag{10}$$

верно при любых значениях $x_1, \ldots, x_n, y$.

Найдем теперь примитивно рекурсивную схему для F. Прежде всего, из (7) и (8) имеем

$$F(x_1, \ldots, x_n, 0) = p_0^{g(x_1, \ldots, x_n)}. \tag{11}$$

В силу (8) имеем также

$$F(x_1, \ldots, x_n, y+1) = F(x_1, \ldots, x_n, y)\, p_{y+1}^{f(x_1, \ldots, x_n, y+1)}.$$

Заменяя здесь терм $f(x_1, \ldots, x_n, y+1)$ его значением из (7), получим

$$F(x_1, \ldots, x_n, y+1) = F(x_1, \ldots, x_n, y) \times$$
$$\times\, p_{y+1}^{h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, \alpha_1(y+1)), \ldots, f(x_1, \ldots, x_n, \alpha_s(y+1)))}. \tag{12}$$

Соотношения (11), (12) можно при помощи формул (10) представить в виде

$$\begin{cases} F(x_1, \ldots, x_n, 0) = G(x_1, \ldots, x_n), \\ F(x_1, \ldots, x_n, y+1) = H(x_1, \ldots, x_n, y, F(x_1, \ldots, x_n, y)), \end{cases}$$

где

$$G(x_1, \ldots, x_n) = p_0^{g(x_1, \ldots, x_n)},$$

$$H(x_1, \ldots, x_n, y, z) =$$
$$= z p_{y+1}^{h(x_1, \ldots, x_n, y, \mathrm{ex}(\alpha_1(y+1), z), \ldots, \mathrm{ex}(\alpha_s(y+1), z))}.$$

Таким образом, функция F получается примитивной рекурсией из функций G и H, а функция f выражается через F согласно (9) формулой

$$f(x_1, \ldots, x_n, y) = \mathrm{ex}(y, F(x_1, \ldots, x_n, y)).$$

Тем самым теорема 1 доказана. В качестве примера обычно приводят последовательность

$$0, 1, 1, 2, 3, 5, 8, 13, 21, \ldots,$$

известную в теории чисел под именем *последовательности Фибоначчи*. Каждый член этой последовательности, начиная с третьего, равен сумме двух предыдущих. Обозначая через $\Phi(n)$ n-й член последовательности Фибоначчи, получим

$$\Phi(0) = 0,$$
$$\Phi(1) = 1,$$
$$\Phi(n+2) = \Phi(n) + \Phi(n+1)$$

или

$$\Phi(0) = 0,$$
$$\Phi(n+1) = \Phi(n) + \Phi(n-1) + \overline{\mathrm{sg}}\, n.$$

Отсюда видно, что функция $\Phi(n)$ возникает возвратной рекурсией из функций

$$g(x) = 0, \quad h(x, y, z_1, z_2) = \overline{\mathrm{sg}}\, y + z_1 + z_2$$

и вспомогательных функций

$$\alpha_1(y) = y \dot{-} 1, \quad \alpha_2(y) = y \dot{-} 2.$$

Так как все эти функции примитивно рекурсивны, то функция Φ также примитивно рекурсивна.

3.3. Нумерация пар и n-ок чисел. Все пары натуральных чисел можно расположить в простую последовательность и притом многими способами. Для определенности рассмотрим следующее расположение этих пар, которое мы будем называть *канторовским*:

$$\langle 0, 0\rangle; \langle 0, 1\rangle, \langle 1, 0\rangle; \langle 0, 2\rangle, \langle 1, 1\rangle, \langle 2, 0\rangle; \langle 0, 3\rangle, \ldots \quad (1)$$

В этой последовательности пары идут в порядке возрастания суммы их членов, а из пар с одинаковой суммой членов ранее идет пара с меньшим первым членом. Обозначим через $\boldsymbol{c}(x, y)$ номер пары $\langle x, y\rangle$ в последовательности (1), причем нумерацию начинаем с нуля. Таким образом,

$$\boldsymbol{c}(0, 0)=0, \ \boldsymbol{c}(0, 1)=1, \ \boldsymbol{c}(1, 0)=2, \ \ldots$$

Число $\boldsymbol{c}(x, y)$ будем называть (канторовским) номером пары $\langle x, y\rangle$. Через $\boldsymbol{l}(n)$ и $\boldsymbol{r}(n)$ обозначим левый и соответственно правый члены пары с номером n. Например,

$$\boldsymbol{l}(2)=1, \ \boldsymbol{r}(2)=0.$$

Мы хотим теперь выразить функции $\boldsymbol{c}$, $\boldsymbol{l}$, $\boldsymbol{r}$ через обычные арифметические функции. Начнем с функции $\boldsymbol{c}$. Пара $\langle x, y\rangle$ находится в отрезке

$$\langle 0, x+y\rangle, \langle 1, x+y-1\rangle, \ldots, \langle x, y\rangle, \ldots, \langle x+y, 0\rangle$$

на x-м месте после пары $\langle 0, x+y\rangle$. Перед парой $\langle 0, x+y\rangle$ в последовательности (1) находятся $x+y$ отрезков, содержащих всего $1+2+\ldots+(x+y)$ пар. Поэтому

$$\boldsymbol{c}(x, y)=\frac{(x+y)(x+y+1)}{2}+x=\frac{(x+y)^2+3x+y}{2}. \qquad (2)$$

Обратно, пусть

$$x=\boldsymbol{l}(n), \quad y=\boldsymbol{r}(n)$$

и, следовательно, $n=\boldsymbol{c}(x, y)$. Из формулы (2) имеем

$$2n=(x+y)^2+3x+y,$$

$$8n+1=(2x+2y+1)^2+8x=(2x+2y+3)^2-8y-8.$$

Отсюда вытекает, что

$$2x+2y+1 \leqslant [\sqrt{8n+1}] < 2x+2y+3,$$

или

$$x+y+1 \leqslant \frac{[\sqrt{8n+1}]+1}{2} < x+y+2.$$

Таким образом,

$$x+y+1=\left[\frac{[\sqrt{8n+1}]+1}{2}\right]. \qquad (3)$$

Сравнивая с формулой (2), получаем

$$\boldsymbol{l}(n) = x = n \dot{-} \frac{1}{2}\left[\frac{[\sqrt{8n+1}]+1}{2}\right]\left[\frac{[\sqrt{8n+1}] \dot{-} 1}{2}\right]. \qquad (4)$$

Из (3) и (4) получаем аналогичное выражение для $\boldsymbol{r}(n)$.

В дальнейшем будет нужен не конкретный вид формул для $\boldsymbol{c}(x, y)$, $\boldsymbol{l}(n)$, $\boldsymbol{r}(n)$, а лишь тот факт, что эти функции выражаются с помощью подстановок через элементарные арифметические функции $+$, $\dot{-}$, $\cdot$, $[\sqrt{\ }]$, $[/2]$.

Отметим, наконец, что из самого определения функций $\boldsymbol{c}(x, y)$, $\boldsymbol{l}(x)$, $\boldsymbol{r}(x)$, как функций, связанных с нумерацией пар, вытекают следующие тождества:

$$\boldsymbol{c}(\boldsymbol{l}(n), \ \boldsymbol{r}(n)) = n, \ \ \boldsymbol{l}(\boldsymbol{c}(x, y)) = x, \ \ \boldsymbol{r}(\boldsymbol{c}(x, y)) = y. \qquad (5)$$

Ясно, что эти тождества не зависят от того, что рассматриваемая нумерация пар канторова. Например, пусть

$$\langle x_0, y_0\rangle, \langle x_1, y_1\rangle, \ldots, \langle x_n, y_n\rangle, \ldots \qquad (6)$$

— какое-нибудь другое расположение совокупности всех пар натуральных чисел в простую последовательность. Обозначая через $C(x, y)$ номер пары $\langle x, y\rangle$ в последовательности (6) и полагая $L(n) = x_n$, $R(n) = y_n$, мы получим снова функции, связанные тождествами

$$C(L(n), \ R(n)) = n,$$
$$L(C(x, y)) = x, \quad R(C(x, y)) = y, \qquad (7)$$

аналогичными тождествам (5). Обратно, если на множестве натуральных чисел определены какие-то функции C, L, R, связанные тождествами (7), то, называя номером пары $\langle x, y\rangle$ число $C(x, y)$, мы получим одно-однозначную нумерацию совокупности натуральных чисел. Связанные с этой нумерацией функции будут как раз функциями C, L, R.

С помощью нумерации пар натуральных чисел легко получить нумерации троек, четверок и т. д. натуральных чисел. С этой целью вводим следующие функции:

$$\boldsymbol{c}^3(x_1, x_2, x_3) = \boldsymbol{c}(\boldsymbol{c}(x_1, x_2), x_3),$$
$$\cdots\cdots\cdots\cdots\cdots \qquad (8)$$
$$\boldsymbol{c}^{n+1}(x_1, x_2, \ldots, x_{n+1}) = \boldsymbol{c}^n(\boldsymbol{c}(x_1, x_2), x_3, \ldots, x_{n+1}),$$

и по определению число $c^n(x_1, \ldots, x_n)$ называем *канторовым номером* n-*ки* $\langle x_1, \ldots, x_n \rangle$. Если

$$c^n(x_1, \ldots, x_n) = x,$$

то из тождеств (5), (8) получаем

$$\begin{aligned} x_n &= \boldsymbol{r}(x), \\ x_{n-1} &= \boldsymbol{rl}(x), \\ &\ldots\ldots\ldots\ldots \\ x_2 &= \boldsymbol{rl} \ldots \boldsymbol{l}(x), \\ x_1 &= \boldsymbol{ll} \ldots \boldsymbol{l}(x). \end{aligned} \tag{9}$$

Вводя обозначения $c_{nn}(x), \ldots, c_{n1}(x)$ для правых частей равенств (9), будем иметь

$$c^n(c_{n1}(x), \ldots, c_{nn}(x)) = x,$$
$$c_{ni}(c^n(x_1, \ldots, x_n)) = x_i \qquad (i = 1, \ldots, n).$$

Это аналоги тождеств (5) для канторовской нумерации n-ок натуральных чисел.

Наконец, можно было бы еще рассмотреть нумерацию совокупности, образованной всеми натуральными числами, всеми парами, всеми тройками и т. д. натуральных чисел. Однако мы этого здесь делать пока не будем, а рассмотрим лишь так называемую функцию Геделя $\boldsymbol{\Gamma}$, заменяющую во многих случаях нумерацию указанной смешанной совокупности.

По определению полагаем

$$\boldsymbol{\Gamma}(x, y) = \mathrm{rest}(\boldsymbol{l}(x), 1 + (y+1)\,\boldsymbol{r}(x)). \tag{10}$$

Отсюда видно, что функция Геделя примитивно рекурсив ная. Основное свойство функции Геделя выражает следующая

Теорема 1. *Какова бы ни была конечная последовательность натуральных чисел* $a_0, a_1, \ldots, a_n$, *система уравнений*

$$\begin{aligned} \boldsymbol{\Gamma}(x, 0) &= a_0, \\ \boldsymbol{\Gamma}(x, 1) &= a_1, \\ &\ldots\ldots \\ \boldsymbol{\Gamma}(x, n) &= a_n \end{aligned} \tag{11}$$

имеет по меньшей мере одно решение x.

Итак, пусть числа $a_0, a_1, \ldots, a_n$ заданы. Будем искать числа u, b такие, чтобы число $x = c(u, b)$ удовлетворяло уравнениям (11). Вследствие соотношения (10) это значит, что нам надо подобрать числа u, b так, чтобы выполнялись соотношения

$$\mathrm{rest}(u, 1+(y+1)b) = a_y \qquad (y = 0, 1, \ldots, n),$$

то есть чтобы выполнялись неравенства

$$1+(y+1)b > a_y \qquad (y = 0, 1, \ldots, n) \tag{12}$$

и чтобы разности $u - a_y$ делились на $1 + (y + 1)b$. Для краткости записей введем обозначения

$$m_y = 1+(y+1)b. \tag{13}$$

Неравенства (12), очевидно, удовлетворятся, если положить

$$b = (1+n+a_0+\ldots+a_n)! \tag{14}$$

Однако такой выбор значения для b гарантирует нам и то важное обстоятельство, что числа $m_0, m_1, \ldots, m_n$ будут попарно взаимно просты. В самом деле, если $0 \leqslant i < j \leqslant n$, то

$$m_j - m_i = (j-i)b \qquad (0 < j-i \leqslant n).$$

Согласно (14), число b делится на $j - i$. Поэтому, если бы числа m_j, m_i имели какой-нибудь общий простой делитель p_s, то на p_s должно было бы делиться и число b. Однако из (13) видно, что m_i и b общих простых делителей не имеют. Таким образом, числа m_j и m_i действительно взаимно простые.

Вводим теперь числа $M_0, M_1, \ldots, M_n$, полагая

$$M_i = m_0 \ldots m_{i-1}m_{i+1} \ldots m_n \qquad (i = 0, 1, \ldots, n).$$

Так как числа $m_0, m_1, \ldots, m_n$ попарно взаимно простые, то числа m_i и M_i также взаимно простые. Поэтому согласно известной теореме арифметики найдутся натуральные числа z_i, w_i, удовлетворяющие соотношению

$$M_i z_i - m_i w_i = 1. \tag{15}$$

Нам оставалось еще подобрать число u так, чтобы каждое число $u - a_y$ делилось на m_y. Покажем теперь, что

этим требованиям заведомо удовлетворяет число

$$u = a_0M_0z_0 + a_1M_1z_1 + \ldots + a_nM_nz_n. \qquad (16)$$

В самом деле, все члены этой суммы делятся на m_i, за исключением члена $a_iM_iz_i$ (так как на m_i делятся M_j для $j \neq i$). Соотношение (15) показывает, что произведение M_iz_i при делении на m_i дает остаток 1. Поэтому при делении на m_i члена $a_iM_iz_i$ получится остаток a_i. Поскольку остальные члены суммы (16) при делении на m_i дают остаток 0, то $\mathrm{rest}\,(u, m_i) = a_i$, что и требовалось.

3.4. Зависимости между операторами примитивной рекурсии и минимизации. Согласно § 2 каждую функцию, примитивно рекурсивную относительно системы функций $\mathfrak{S}$, можно получить из функций $\mathfrak{S}$ и простейших функций $\boldsymbol{o}$, $\boldsymbol{s}$, I_m^n операциями подстановки и примитивной рекурсии. Однако, если исходить не только из простейших функций, а и из нумерационных функций $\boldsymbol{c}$, $\boldsymbol{l}$, $\boldsymbol{r}$, то для получения примитивно рекурсивных относительно $\mathfrak{S}$ функций нет необходимости пользоваться операциями примитивной рекурсии в их общем виде. Достаточно применять лишь некоторый весьма специальный вид этих операций.

Аналогичное положение имеет место и для частично рекурсивных относительно $\mathfrak{S}$ функций. Чтобы получить эти функции из функций системы $\mathfrak{S}$ и функций $\boldsymbol{o}$, $\boldsymbol{s}$, I_m^n, $\boldsymbol{c}$, $\boldsymbol{l}$, $\boldsymbol{r}$, нет нужды пользоваться операциями примитивной рекурсии и минимизации в общем виде, а достаточно употреблять лишь некоторые частные виды этих операций. Более детально эти вопросы будут изучены в пп. 3.5 и 5.4, а пока мы докажем лишь несколько предварительных теорем.

Теорема 1. *Частичная функция* $f(x_1, \ldots, x_n, y)$, *возникающая из частичных функций* $g(x_1, \ldots, x_n)$ *и* $h(x_1, \ldots, x_n, y, z)$ *примитивной рекурсией*:

$$\begin{aligned} &f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n), \\ &f(x_1, \ldots, x_n, y+1) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y)), \end{aligned} \qquad (1)$$

может быть получена специальной рекурсией вида

$$\begin{aligned} &\varphi(x, 0) = x, \\ &\varphi(x, y+1) = \Phi(\varphi(x, y)) \end{aligned} \qquad (2)$$

и подстановками из функций g, h, $\boldsymbol{c}$, $\boldsymbol{l}$, $\boldsymbol{r}$, $\boldsymbol{o}$, $\boldsymbol{s}$, I_m^n.

В частности, каждая функция, примитивно рекурсивная относительно системы $\mathfrak{S}$ частичных функций, может быть получена из функций $\mathfrak{S}$ и функций $\boldsymbol{c}$, $\boldsymbol{l}$, $\boldsymbol{r}$, $\boldsymbol{o}$, $\boldsymbol{s}$, I^n_m *операциями подстановки и специальными рекурсиями вида* (2).

Вводим функцию

$$\psi(t, y) = c^{n+2}(c_{n1}(t), \ldots, c_{nn}(t), y, f(c_{n1}(t), \ldots, c_{nn}(t), y)), \tag{3}$$

где c^{n+2}, c_{ni} — нумерационные функции, определенные в предыдущем разделе и получающиеся подстановками из функций $\boldsymbol{c}$, $\boldsymbol{l}$, $\boldsymbol{r}$. Из соотношения (3) непосредственно следует

$$f(x_1, \ldots, x_n, y) = c_{n+2\,n+2}(\psi(c^n(x_1, \ldots, x_n), y)). \tag{4}$$

Первое из равенств (1) дает

$$\psi(t, 0) = c^{n+2}(c_{n1}(t), \ldots, c_{nn}(t), 0, g(c_{n1}(t), \ldots, c_{nn}(t))). \tag{5}$$

Из (3) и (1) непосредственно следует

$$\begin{gathered}\psi(t, y+1) = c^{n+2}(c_{n1}(t), \ldots, c_{nn}(t), y+1,\\ h(c_{n1}(t), \ldots, c_{nn}(t), y, c_{n+2\,n+2}(\psi(t, y)))).\end{gathered} \tag{6}$$

Так как

$$c_{ni}(t) = c_{n+2\,i}(\psi(t, y)), \quad y = c_{n+2\,n+1}(\psi(t, y)),$$

то равенство (6) можно переписать в виде

$$\psi(t, y+1) = \Phi(\psi(t, y)), \tag{7}$$

где

$$\begin{gathered}\Phi(u) = c^{n+2}(c_{n+2\,1}(u), \ldots, c_{n+2\,n+1}(u)+1,\\ h(c_{n+2\,1}(u), \ldots, c_{n+2\,n+2}(u))).\end{gathered}$$

Итак, функция f получается из функций g, h, $\boldsymbol{l}$, $\boldsymbol{r}$, $\boldsymbol{c}$, $\boldsymbol{o}$, $\boldsymbol{s}$, I^n_m подстановками и рекурсией

$$\begin{gathered}\psi(x, 0) = G(x),\\ \psi(x, y+1) = \Phi(\psi(x, y)).\end{gathered}$$

Вместо функции ψ вводим теперь функцию φ посредством рекурсии

$$\varphi(x, 0) = x, \quad \varphi(x, y+1) = \Phi(\varphi(x, y)),$$

имеющей требуемый вид (2). Так как

$$\psi(x, y) = \Phi(\Phi(\ldots \Phi(G(x)) \ldots)),$$
$$\varphi(x, y) = \Phi(\Phi(\ldots \Phi(x) \ldots)),$$

то

$$\psi(x, y) = \varphi(G(x), y)$$

и, следовательно, функция f получается из функций $\boldsymbol{c}, \boldsymbol{l}, \boldsymbol{r}, \boldsymbol{o}, \boldsymbol{s}, I^n_m, g, h$ подстановками и рекурсией вида (2).

Полагая в теореме 1 $n = 0$, $g = 0$, получим следующий частный случай: функция $f(y)$, возникающая из функции $h(y, z)$ рекурсией

$$f(0) = 0, \quad f(y+1) = h(y, f(y)),$$

может быть получена из функций $h, \boldsymbol{c}, \boldsymbol{l}, \boldsymbol{r}, \boldsymbol{o}, \boldsymbol{s}$ подстановками и рекурсией вида

$$\varphi(0) = 0, \quad \varphi(y+1) = \Phi(\varphi(y)),$$

где Φ — подходящая суперпозиция функций h, $\boldsymbol{c}$, $\boldsymbol{l}$, $\boldsymbol{r}$, $\boldsymbol{o}$, $\boldsymbol{s}$, I^n_m.

Теорема 2. *Частичная функция* $f(x_1, \ldots, x_n)$, *получающаяся из частичной функции* $g(x_1, \ldots, x_n, y)$ *операцией минимизации*

$$f(x_1, \ldots, x_n) = \mu_y(g(x_1, \ldots, x_n, y) = 0), \tag{8}$$

может быть получена из функций g, $\boldsymbol{c}$, $\boldsymbol{l}$, $\boldsymbol{r}$ *подстановками и операцией минимизации специального вида*

$$F(x) = \mu_y(G(x, y) = 0). \tag{9}$$

Действительно, вводя функцию

$$F(x) = f(\boldsymbol{c}_{n1}(x), \ldots, \boldsymbol{c}_{nn}(x)),$$

получим

$$f(x_1, \ldots, x_n) = F(\boldsymbol{c}^n(x_1, \ldots, x_n)).$$

В то же время из (8) вытекает

$$F(x) = \mu_y(G(x, y) = 0),$$

где

$$G(x, y) = g(\boldsymbol{c}_{n1}(x), \ldots, \boldsymbol{c}_{nn}(x), y).$$

Из теорем 1 и 2 непосредственно получаем

Следствие. *Каждая функция, частично рекурсивная относительно системы частичных функций $\mathfrak{S}$, может быть получена из функций системы $\mathfrak{S}$ и функций $\boldsymbol{c}, \boldsymbol{l}, \boldsymbol{r}, \boldsymbol{o}, \boldsymbol{s}$, I_m^n конечным числом операций подстановки, специальных рекурсий вида* (2) *и специальных минимизаций вида* (9).

Столь же просто доказывается и

Теорема 3 (Гедель [14]). *Функция $\varphi(x_1, \ldots, x_n, y)$, получающаяся примитивной рекурсией* (1) *из всюду определенных функций $g(x_1, \ldots, x_n)$ и $h(x_1, \ldots, x_n, y, z)$, может быть получена из этих функций и функций $\boldsymbol{\Gamma}$, $\boldsymbol{c}$, $\boldsymbol{l}$, $\boldsymbol{r}$, $+$, $\dot{-}$, $\boldsymbol{s}$, $\boldsymbol{o}$, I_k^m конечным числом подстановок и минимизаций.*

Ввиду теоремы 1 достаточно провести доказательство лишь для случая, когда $n = 1$ и рекурсия имеет специальный вид (2). Полагая x, y произвольно заданными числами, рассмотрим систему уравнений относительно z:

$$\begin{aligned} \boldsymbol{\Gamma}(z, 0) &= \varphi(x, 0), \\ \boldsymbol{\Gamma}(z, 1) &= \varphi(x, 1), \\ &\ldots\ldots\ldots \\ \boldsymbol{\Gamma}(z, y) &= \varphi(x, y), \end{aligned} \tag{10}$$

где $\boldsymbol{\Gamma}(z, y)$ — функция Геделя из п. 3.3. В силу (2) эту систему можно переписать в виде

$$\begin{aligned} \boldsymbol{\Gamma}(z, 0) &= x, \\ \boldsymbol{\Gamma}(z, u+1) &= \Phi(\boldsymbol{\Gamma}(z, u)) \quad (0 \leqslant u < y) \end{aligned}$$

или

$$\begin{aligned} \boldsymbol{\Gamma}(z, 0) &= x, \\ \mu_u(|y-u|\,\overline{\mathrm{sg}}\,|\boldsymbol{\Gamma}(z, u+1) - \Phi(\boldsymbol{\Gamma}(z, u))| = 0) &= y. \end{aligned} \tag{11}$$

Множитель $|y-u|$ введен здесь для того, чтобы функция

$$F(y, z) = \mu_u(|y-u|\,\overline{\mathrm{sg}}\,|\boldsymbol{\Gamma}(z, u+1) - \Phi(\boldsymbol{\Gamma}(z, u))| = 0)$$

была всюду определенной. Согласно основному свойству функции $\boldsymbol{\Gamma}$ система (10), а значит, и равносильные ей система (11) и уравнение

$$|\boldsymbol{\Gamma}(z, 0) - x| + |F(y, z) - y| = 0 \tag{12}$$

имеют решение z для каждых x, y. Так как левая часть

уравнения (12) является всюду определенной функцией, то формула

$$z = \mu_u (|\mathbf{\Gamma}(u,\ 0) - x| + |F(y,\ u) - y| = 0)$$

дает наименьшее решение z системы (10). Обозначая это решение через $G(x,\ y)$, из последнего уравнения системы (10) получим

$$\varphi(x,\ y) = \mathbf{\Gamma}(G(x,\ y),\ y).$$

Таким образом, функции F, G и φ получаются подстановками и минимизациями из функции Φ и функций $\mathbf{\Gamma}$, $+$, $\dot{-}$, I_k^m. Дополнительные функции $\boldsymbol{c}, \boldsymbol{l}, \boldsymbol{r}, \boldsymbol{s}, \boldsymbol{o}$ указаны в теореме 3 для того, чтобы можно было применить теорему 1 и свести дело к случаю рекурсии специального вида (2). С другой стороны, с их помощью легко получаются использованные функции $x \cdot y$ и $\overline{\mathrm{sg}}\, x$. Именно,

$$\overline{\mathrm{sg}}\, x = 1 \dot{-} x = \boldsymbol{so}(x) \dot{-} x = \boldsymbol{so}(x) \dot{-} I_1^1(x),$$

$$[y/2] = \mu_z (y \dot{-} (2z + 1) = 0),$$

$$\boldsymbol{c}(x,\ x) = 2x^2 + 2x,\ 2x^2 = \boldsymbol{c}(x,\ x) \dot{-} 2x,\ x^2 = [2x^2/2],$$

$$x \cdot y = \left[\frac{((x+y)^2 \dot{-} x^2) \dot{-} y^2}{2} \right].$$

Согласно п. 2.4 функция называется общерекурсивной, если она может быть получена из простейших функций операциями подстановки, примитивной рекурсии и операциями минимизации, не выводящими за пределы всюду определенных функций. Поэтому из теоремы 3 следует, что *каждая общерекурсивная функция может быть получена из функций* $\mathbf{\Gamma}$, $\boldsymbol{c}$, $\boldsymbol{l}$, $\boldsymbol{r}$, $\boldsymbol{s}$, $\boldsymbol{o}$, $+$, $\dot{-}$, I_k^m *конечным числом подстановок и специальных операций минимизации вида* (9), *применяемых лишь к таким функциям, для которых результатом минимизации служит всюду определенная функция.*

Совокупность исходных функций здесь взята с некоторым запасом. Минимальные совокупности будут определены в пп. 3.5 и 5.4.

3.5. Одноместные примитивно рекурсивные функции. Указанное в п. 2.2 основное определение примитивно рекурсивных функций имеет смысл лишь тогда, когда рассматриваются одновременно функции от любого числа аргу-

ментов. Однако довольно часто встречается положение, когда достаточно рассматривать лишь функции одной переменной. Поэтому естественно возникает вопрос: нельзя ли найти такое определение понятия примитивной рекурсивности одноместных функций, которое выражалось бы целиком в терминах теории одноместных функций? Доказываемая ниже теорема Робинсона дает положительный ответ на этот вопрос. Получается она надлежащим усилением теоремы 1 из предыдущего раздела.

Рассмотрим следующие операции над одноместными частичными функциями:

а) Двуместная операция сложения одноместных частичных функций. Эта операция обозначается символом $+$ и определяется формулой

$$(f+g)(x)=f(x)+g(x).$$

б) Двуместная операция композиции одноместных частичных функций. Эта операция обозначается далее символом $*$ и определяется соотношением

$$(f*g)(x)=f(g(x)).$$

в) Одноместная операция итерирования одноместной частичной функции. Эта операция далее обозначается символом $\boldsymbol{J}$ и определяется следующим образом: говорят, что одноместная частичная функция f возникает операцией итерирования из одноместной частичной функции g (символически $f = \boldsymbol{J}g = g^{\boldsymbol{J}}$), если

$$f(0)=0,$$
$$f(n+1)=g(f(n)).$$

Конечно, операции $+$, $*$, $\boldsymbol{J}$ легко выразить через прежние операции. Например, ясно, что

$$f+g=\boldsymbol{S}^3(+,\ f,\ g),$$
$$f*g=\boldsymbol{S}^2(f,\ g),$$
$$\boldsymbol{J}g=\boldsymbol{R}(\boldsymbol{o},\ \boldsymbol{S}^2(I_2^2,\ g)).$$

Характерной особенностью новых операций служит то, что они выполнимы над одноместными функциями и дают в результате снова одноместные функции.

Теорема (Робинсон [89]). *Все одноместные примитивно рекурсивные функции и только они могут быть получены из функций* $\boldsymbol{s}(x) = x + 1$ и $\boldsymbol{q}(x) = x \dot{-} [\sqrt{x}]^2$ *конечным числом операций сложения, композиции и итерирования функций.*

Условимся временно называть одноместную функцию f допустимой, если она может быть получена из функций $\boldsymbol{s}$ и $\boldsymbol{q}$ конечным числом операций $+, *, \boldsymbol{J}$. Нам надо доказать два утверждения:

1) *все допустимые одноместные функции примитивно рекурсивны;*

2) *все примитивно рекурсивные одноместные функции допустимы.*

Первое утверждение очевидно, так как исходные функции $\boldsymbol{s}$ и $\boldsymbol{q}$ примитивно рекурсивны, а операции сложения, композиции и итерирования, примененные к примитивно рекурсивным функциям, дают снова примитивно рекурсивные функции. Поэтому дело сводится к доказательству второго утверждения.

Прежде всего распространим понятие допустимости и на многоместные функции, называя n-местную функцию $f(x_1, \ldots, x_n)$ $(n \geqslant 2)$ допустимой, если для любых одноместных допустимых функций $\alpha_1(t), \ldots, \alpha_n(t)$ одноместная функция $f(\alpha_1(t), \ldots, \alpha_n(t))$ является допустимой. А теперь вместо утверждения 2) будем доказывать более общее утверждение:

3) *все* (одноместные и многоместные) *примитивно рекурсивные функции допустимы.*

Доказательство разобьем на ряд вспомогательных утверждений, из сопоставления которых и получится утверждение 3).

А) *Суперпозиция и сумма допустимых функций* (*с любым числом аргументов*) *суть функции допустимые.*

В самом деле, пусть $g(x_1, \ldots, x_n)$, $h_1(x_1, \ldots, x_m), \ldots, h_n(x_1, \ldots, x_m)$ допустимы и

$$f(x_1, \ldots, x_m) = g(h_1(x_1, \ldots, x_m), \ldots, h_n(x_1 \ldots, x_m)).$$

Тогда для любых допустимых функций $\alpha_1(t), \ldots, \alpha_m(t)$ одноместная функция

$$f(\alpha_1(t), \ldots, \alpha_m(t)) = g(\beta_1(t), \ldots, \beta_n(t)),$$

где 65

$$\beta_i(t) = h_i(\alpha_1(t), \ldots, \alpha_m(t)) \quad (i = 1, \ldots, n),$$

будет допустимой, так как по условию функции g, h_i допустимы. Но тогда допустима и функция f. Аналогично доказывается и допустимость суммы допустимых функций.

Б) *Одноместные функции* $\boldsymbol{o}$, I_1^1, sg, $\overline{\text{sg}}$ *и многоместные функции* I_m^n, $ax + by + c$, *где* a, b, c — *фиксированные натуральные числа, являются допустимыми.*

Допустимость функций I_1^1, $\boldsymbol{o}$, sg, $\overline{\text{sg}}$ вытекает из очевидных соотношений

$$\boldsymbol{s}^J = I_1^1, \quad I_1^{1J} = \boldsymbol{o},$$

$$\text{sg} = (\boldsymbol{s} * \boldsymbol{o})^J, \quad \overline{\text{sg}}\, x = \boldsymbol{q}(2 + 2\,\text{sg}\, x).$$

Так как для допустимых функций $\alpha_1(t), \ldots, \alpha_n(t)$ функция

$$I_m^n(\alpha_1(t), \ldots, \alpha_n(t)) = \alpha_m(t)$$

допустимая, то I_m^n допустима. Наконец, из формулы

$$ax + by + c = I_1^2(x, y) + \ldots + I_2^2(x, y) + \ldots + 1 + \ldots + 1$$

и допустимости суммы заключаем, что функция $ax + by + c$ также допустимая.

В) *Функция* x^2 *допустимая.* В силу Б) функция

$$\overline{\text{sg}}\, \boldsymbol{q}(x) = \begin{cases} 1, & \text{если } x \text{ — полный квадрат}, \\ 0, & \text{если } x \text{ — неполный квадрат}, \end{cases}$$

допустимая. С другой стороны, равенства $0^2 = 0$ и

$$(x+1)^2 = x^2 + 2x + 1 = \varphi(x^2) + 1,$$

где

$$\varphi(t) = t + 2\left[\sqrt{t}\right],$$

показывают, что x^2 есть итерация функции $\varphi + 1$. В свою

очередь, равенства $\varphi(0)=0$ и

$$\varphi(t+1)=\varphi(t)+1+2\left(\left[\sqrt{t+1}\right]-\left[\sqrt{t}\right]\right)=$$
$$=\varphi(t)+1+2\,\overline{\mathrm{sg}}\,q\,(\varphi(t)+4)$$

показывают, что φ получается итерированием функции

$$\psi(u)=u+1+2\,\overline{\mathrm{sg}}\,q\,(u+4).$$

Так как последняя функция допустимая, то допустимы функции φ и x^2.

Г) *Функции* $[x/2]$, $\left[\sqrt{x}\right]$, xy, $x \dot{-} y$ *допустимые*. Из А), Б), В) следует, что функция $q\,((x+y)^2+5x+3y+4)$ допустимая. Так как

$$(x+y)^2+5x+3y+4=(x+y+2)^2+x-y,$$

то для $x \geqslant y$ имеем

$$(x+y+2)^2 \leqslant (x+y+2)^2+x-y<(x+y+3)^2$$

и, следовательно,

$$q\,((x+y)^2+5x+3y+4)=x-y \quad (x \geqslant y).$$

Если же $x<y$, то из

$$(x+y+1)^2<(x+y+1)^2+3x+y+3=$$
$$=(x+y+2)^2+x-y<(x+y+2)^2$$

вытекает, что

$$q\,((x+y)^2+5x+3y+4)=3x+y+3 \quad (x<y).$$

Вводя для $q\,((x+y)^2+5x+3y+4)$ обозначение $x \div y$, приходим к заключению, что функция $x \div y$, определяемая схемой

$$x \div y=\begin{cases} x-y, & \text{если } x \geqslant y, \\ 3x+y+3, & \text{если } x<y, \end{cases}$$

является допустимой.

Согласно определению итерирования, имеем

$$\overline{\mathrm{sg}}^J(0)=0,$$
$$\overline{\mathrm{sg}}^J(1)=\overline{\mathrm{sg}}\,0=1,$$
$$\overline{\mathrm{sg}}^J(2)=\overline{\mathrm{sg}}\,1=0,$$
$$\dots\dots\dots\dots$$

то есть

$$\overline{\mathrm{sg}}^J(x) = \begin{cases} 0, & \text{если } x \text{ четное,} \\ 1, & \text{если } x \text{ нечетное.} \end{cases}$$

Рассмотрим функцию

$$f(x) = x^2 + [x/2].$$

Так как

$$f(0) = 0, \quad \left[\sqrt{f(x)}\right] = x$$

и

$$\overline{\mathrm{sg}}^J x = \overline{\mathrm{sg}}^J x^2 = \overline{\mathrm{sg}}^J \left[\sqrt{f(x)}\right],$$

то из

$$f(x+1) = f(x) + 2x + 1 + \left(\left[\frac{x+1}{2}\right] - \left[\frac{x}{2}\right]\right) =$$

$$= f(x) + 2\left[\sqrt{f(x)}\right] + 1 + \overline{\mathrm{sg}}^J \left[\sqrt{f(x)}\right]^2 = F(f(x)),$$

где

$$F(t) = t + 2\left[\sqrt{t}\right] + 1 + \overline{\mathrm{sg}}^J \left[\sqrt{t}\right]^2,$$

следует, что f получается итерированием функции F. Согласно В) функция $t + 2\left[\sqrt{t}\right]$ допустима. С другой стороны,

$$\left[\sqrt{t}\right]^2 = t - q(t) = t \dot{-} q(t)$$

и потому функция $\left[\sqrt{t}\right]^2$ также допустима. Но тогда допустима и функция $F(t)$, а вместе с нею будут допустимы и функции

$$f(x) = x^2 + \left[\frac{x}{2}\right], \quad \left[\frac{x}{2}\right] = f(x) \dot{-} x^2.$$

Наконец, из допустимости функций $\left[\frac{x}{2}\right]$ и $\varphi(t) = t + 2\left[\sqrt{t}\right]$ вытекает допустимость функций

$$2\left[\sqrt{t}\right] = \varphi(t) \dot{-} t, \quad \left[\sqrt{t}\right] = \left[\frac{2\left[\sqrt{t}\right]}{2}\right]$$

и

$$xy = \frac{((x+y)^2 \dot{-} x^2) \dot{-} y^2}{2}.$$

Остается доказать допустимость функции $\dot{-}$. Для этого заметим сначала, что

$$((x \div y) + y) \div x = \begin{cases} 0, & \text{если } x \geqslant y, \\ 2x + 2y + 3, & \text{если } x < y, \end{cases}$$

и, значит, функция

$$\beta(x, y) = \overline{\mathrm{sg}}\,(((x \div y) + y) \div x) = \begin{cases} 1, & \text{если } x \geqslant y, \\ 0, & \text{если } x < y, \end{cases}$$

допустимая. Но тогда допустима и функция

$$x \dot{-} y = (x \div y)\,\beta(x, y).$$

Д) *Если* $g(x_1, \ldots, x_n)$ *и* $h(x_1, \ldots, x_n, y, z)$ — *допустимые функции, то функция* f, *возникающая из* g *и* h *примитивной рекурсией*:

$$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n),$$
$$f(x_1, \ldots, x_n, y + 1) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y)),$$

также допустимая.

Согласно теореме 1 предыдущего раздела функция f может быть получена из функций g, h, $\boldsymbol{c}$, $\boldsymbol{l}$, $\boldsymbol{r}$, $\boldsymbol{o}$, $\boldsymbol{s}$, I_m^n подстановками и рекурсией вида

$$\varphi(x, 0) = x, \quad \varphi(x, y + 1) = \Phi(\varphi(x, y)), \tag{1}$$

где $\Phi(z)$ — подходящая суперпозиция функций g, h, $\boldsymbol{o}$, $\boldsymbol{s}$, I_m^n, $\boldsymbol{c}$, $\boldsymbol{l}$, $\boldsymbol{r}$. Из ранее доказанного следует, что функции $\boldsymbol{c}$, $\boldsymbol{l}$, $\boldsymbol{r}$, $\boldsymbol{o}$, $\boldsymbol{s}$, I_m^n допустимые и что суперпозиция допустимых функций допустимая. Поэтому нам достаточно показать, что если функция $\Phi(z)$ допустимая, то функция $\varphi(x, y)$, связанная с Φ рекурсией (1), также допустимая.

Рассмотрим вспомогательные функции

$$v(x) = \boldsymbol{q}([\sqrt{x}]), \quad w(x, y) = ((x + y)^2 + y)^2 + x. \tag{2}$$

Из неравенств

$$((x + y)^2 + y)^2 \leqslant ((x + y)^2 + y)^2 + x < ((x + y)^2 + y + 1)^2,$$
$$(x + y)^2 \leqslant (x + y)^2 + y < (x + y + 1)^2$$

вытекает, что

$$\boldsymbol{q}(w(x, y)) = x, \quad v(w(x, y)) = y. \tag{3}$$

Покажем, что если для некоторого x окажется $q(x+1) \neq 0$, то

$$q(x+1) = q(x)+1, \quad v(x+1) = v(x). \tag{4}$$

В самом деле, условие $q(x+1) \neq 0$ означает, что $x+1$ не есть полный квадрат. Но тогда расстояние от ближайшего меньшего полного квадрата до $x+1$ на 1 больше расстояния от этого квадрата до x, то есть $q(x+1) = q(x)+1$. С другой стороны, так как $x+1$ не есть полный квадрат, то $[\sqrt{x+1}] = [\sqrt{x}]$ и потому $v(x+1) = v(x)$.

Теперь мы вместо функции $\varphi(x, y)$, определяемой рекурсией (1), введем функцию

$$\theta(x) = \varphi(v(x),\ q(x)).$$

Из соотношений (3) получаем

$$\theta(w(y, x)) = \varphi(x, y).$$

Таким образом, функция $\varphi(x, y)$ заведомо допустимая, если допустима функция $\theta(x)$. Посмотрим, какой рекурсивной схеме удовлетворяет $\theta(x)$. Формулы (1), (2) дают $\theta(0) = 0$.

Далее, если $q(x+1) \neq 0$, то в силу (4) имеем

$$\theta(x+1) = \varphi(v(x), \quad q(x)+1) = \Phi(\theta(x)).$$

Если же $q(x+1) = 0$, то

$$\theta(x+1) = \varphi(v(x+1),\ 0) = v(x+1).$$

Поэтому для любых значений x имеем

$$\theta(x+1) = \Phi(\theta(x))\,\mathrm{sg}(q(x+1)) + v(x+1)\,\overline{\mathrm{sg}}(q(x+1)).$$

Итак, вводя допустимую функцию

$$\Theta(x, z) = \Phi(z)\,\mathrm{sg}(q(x+1)) + v(x+1)\,\overline{\mathrm{sg}}(q(x+1)),$$

мы можем рекурсивную схему для $\theta(x)$ представить в виде

$$\theta(0) = 0, \quad \theta(x+1) = \Theta(x, \theta(x)). \tag{5}$$

Согласно частному случаю теоремы 1 (п. 3.4) функция $\theta(x)$, удовлетворяющая схеме (5), может быть получена из допустимых функций Θ, $\boldsymbol{c}$, $\boldsymbol{l}$, $\boldsymbol{r}$, $\boldsymbol{o}$, $\boldsymbol{s}$, I^n_m подстановками и итерированием подходящей суперпозиции функций

Θ, $\boldsymbol{c}$, $\boldsymbol{l}$, $\boldsymbol{r}$, $\boldsymbol{o}$, $\boldsymbol{s}$, I^n_m. Таким образом, функция $\theta(x)$ допустимая и утверждение Д) доказано.

Из утверждений А), Б), Д) непосредственно вытекает, что *все примитивно рекурсивные функции допустимые.*

В самом деле, каждая примитивно рекурсивная функция может быть получена из простейших функций $\boldsymbol{o}$, $\boldsymbol{s}$, I^n_m конечным числом операций подстановки и примитивной рекурсии. Согласно Б) простейшие функции допустимы. Согласно А), Д) подстановки и примитивные рекурсии допустимых функций суть функции допустимые. Поэтому все примитивно рекурсивные функции допустимы и теорема доказана.

Теорема Робинсона может быть изящно сформулирована на языке теории алгебр. Обозначим через $\mathfrak{F}^1$ и $\mathfrak{F}^1_{\text{pr. r}}$ совокупность всех одноместных частичных функций и соответственно совокупность всех одноместных примитивно рекурсивных функций. Алгебры

$$\mathfrak{P} = \langle \mathfrak{F}^1;\ +,\ *,\ \boldsymbol{J} \rangle,$$

$$\mathfrak{P}_{\text{pr. r}} = \langle \mathfrak{F}^1_{\text{pr. r}};\ +,\ *,\ \boldsymbol{J} \rangle$$

называются *алгебрами Робинсона*. Теорема Робинсона означает, что совокупность $\mathfrak{F}^1_{\text{pr. r}}$ порождается функциями $\boldsymbol{s}$, $\boldsymbol{q}$ с помощью операций $+$, $*$, $\boldsymbol{J}$, или, что то же самое, что подалгебра $\mathfrak{P}^1_{\text{pr. r}}$ порождается в $\mathfrak{P}$ элементами $\boldsymbol{s}$ и $\boldsymbol{q}$.

Теорема Робинсона была сформулирована и доказана выше лишь для примитивно рекурсивных функций. Однако верна и более общая аналогичная теорема о функциях, примитивно рекурсивных относительно какой-нибудь фиксированной системы частичных функций.

Обобщенная теорема Робинсона. *Те и только те частичные одноместные функции примитивно рекурсивны относительно произвольно заданной системы $\mathfrak{S}$ одноместных частичных функций, которые можно получить из функций системы $\mathfrak{S}$ и функций $\boldsymbol{s}$, $\boldsymbol{q}$ конечным числом операций $+$, $*$, $\boldsymbol{J}$.*

Доказательство, изложенное выше для теоремы Робинсона, остается верным и для обобщенной теоремы Робинсона, если в нем всюду слово *функция* заменить словами *частичная функция*, а слова *примитивно рекурсивная* заменить словами *примитивно рекурсивная относительно* $\mathfrak{S}$.

На языке теории алгебр обобщенную теорему Робинсона можно, очевидно, сформулировать следующим образом: *совокупность всех одноместных частичных функций, примитивно рекурсивных относительно системы одноместных частичных функций* $\mathfrak{S}$, *является совокупностью, порожденной системой* $\mathfrak{S}$ *и функциями* $\boldsymbol{s}$, $\boldsymbol{q}$ *с помощью операций* $+$, $*$, $\boldsymbol{J}$.

Дополнения, примеры и задачи

1. Доказать примитивную рекурсивность следующих функций: а) функции $\sigma(n)$, равной сумме всех делителей натурального числа n; б) функции Эйлера $\varphi(n)$, равной числу натуральных чисел, не превосходящих n и взаимно простых с n.

2. Доказать, что абсолютная величина многочлена от x с целыми коэффициентами есть примитивно рекурсивная функция от x.

3. Пусть $\sqrt{2} = a_0, a_1 a_2 \ldots a_n \ldots$ — разложение числа $\sqrt{2}$ в бесконечную десятичную дробь. Показать, что a_n есть примитивно рекурсивная функция от n.

4. Показать, что $[x\sqrt{2}]$, $[e^x]$ суть примитивно рекурсивные функции от x.

5. Показать, что функции f_1, f_2, определяемые посредством совместной рекурсии вида

$$f_1(0) = a_1, \qquad f_2(0) = a_2,$$
$$f_1(x+1) = h_1(x, f_1(x), f_2(x)), \quad f_2(x+1) = h_2(x, f_1(x), f_2(x))$$

с помощью заданных функций h_1, h_2, являются примитивно рекурсивными относительно h_1, h_2. Определить понятие совместной возвратной рекурсии и доказать аналогичное утверждение для нее.

6. Располагаем все пары натуральных чисел в следующем порядке:

$$\langle 0, 0\rangle; \langle 0, 1\rangle, \langle 1, 0\rangle, \langle 1, 1\rangle; \langle 0, 2\rangle, \langle 2, 0\rangle, \langle 1, 2\rangle, \langle 2, 1\rangle, \langle 2, 2\rangle; \ldots$$

Доказать, что номер $C(x, y) = z$ пары $\langle x, y\rangle$ в этой последовательности и соответствующие функции $x = L(z)$, $y = R(z)$ примитивно рекурсивные.

7. Обозначим через $\mathfrak{F}^2$ совокупность всех частичных функций двух переменных. На $\mathfrak{F}^2$ определяем операции $+$, $*$, $\boldsymbol{J}$ посредством формул:

$$(f+g)(x, y) = f(x, y) + g(x, y),$$
$$(f*g)(x, y) = f(x, g(x, y)),$$
$$f^J(x, 0) = 0, \quad f^J(x, y+1) = f(y, f^J(x, y)).$$

Доказать следующий аналог теоремы Робинсона: двуместная функция f тогда и только тогда примитивно рекурсивна относительно произвольной системы двуместных частичных функций $\mathfrak{S}$,

когда f может быть получена конечным числом операций $+$, $*$, J из функций системы $\mathfrak{S}$ и функций I_1^2, I_2^2, $s(I_2^2)$, $q(I_2^2)$.

8. Согласно теореме Робинсона алгебра $\mathfrak{P}_{\mathrm{pr.\,r}} = \langle \mathfrak{F}^1_{\mathrm{pr.\,r}}; +, *, J \rangle$ порождается парой функций s, q. Показать, что алгебра $\mathfrak{P}_{\mathrm{pr.\,r}}$ порождается и парой функций s, l (также парой s, r), где l, r — канторовы функции из п. 3.3 (см. Р. Робинсон [89]).

9. Вводим бинарную операцию $\vee$ над одноместными функциями, полагая по определению $(f \vee g)(x) = c(f(x), g(x))$.

Показать, что алгебра $\langle \mathfrak{F}^1_{\mathrm{pr.\,r}}; \vee, *, J \rangle$ порождается парой функций s, l (и парой r, s) (Р. Робинсон [89]).

10. Алгебра $\langle \mathfrak{F}^1_{\mathrm{pr.\,r}}; *, J \rangle$ порождается парой подходящих функций и не порождается никакой одной функцией из $\mathfrak{F}^1_{\mathrm{pr.\,r}}$ (Ю. Робинсон [87]).

§ 4. РЕКУРСИВНО ПЕРЕЧИСЛИМЫЕ МНОЖЕСТВА

До сих пор понятия примитивной рекурсивности, общей и частичной рекурсивности были определены лишь для функций. Теперь эти понятия будут перенесены на подмножества натуральных чисел, а также на множества некоторых других объектов. В данном параграфе изучаются лишь простейшие свойства этих новых понятий. Более тонкие свойства будут изучаться в последующих разделах параллельно с изучением свойств функций.

4.1. Рекурсивные и примитивно рекурсивные множества. Подмножество A множества всех натуральных чисел N называется *рекурсивным* (*примитивно рекурсивным*), если характеристическая функция множества A частично рекурсивна (соответственно примитивно рекурсивна).

Так как все примитивно рекурсивные функции частично рекурсивны, то *каждое примитивно рекурсивное множество рекурсивно*. Обратное неверно: в п. 5.2 будут построены рекурсивные множества, не являющиеся примитивно рекурсивными.

С точки зрения теории алгоритмов из двух введенных понятий — рекурсивного множества и примитивно рекурсивного множества — основным следует считать первое благодаря следующему обстоятельству.

Проблемой вхождения для числового множества A называется задача отыскания алгоритма, который по стандартной записи (например, десятичной) произвольного натурального числа a позволяет узнать, входит число a в A

или нет, то есть позволяет вычислять значения характеристической функции множества $\boldsymbol{A}$. В силу тезиса Чёрча существование такого алгоритма равносильно рекурсивности характеристической функции. Поэтому можно сказать, что рекурсивные множества — это множества с алгоритмически разрешимой проблемой вхождения.

Выведем теперь несколько основных свойств рекурсивных и примитивно рекурсивных множеств.

Характеристическими функциями пустого множества $\varnothing$ и множества всех натуральных чисел $\boldsymbol{N}$ являются постоянные одноместные функции 1 и 0. Эти функции примитивно рекурсивные. Поэтому множества $\varnothing$ и $\boldsymbol{N}$ также примитивно рекурсивные.

Характеристической функцией для конечного множества чисел $\{a_1, \ldots, a_n\}$ $(a_1 < a_2 < \ldots < a_n)$ служит примитивно рекурсивная функция

$$\mathrm{sg}\,(|x-a_1|\cdot|x-a_2|\cdot\ldots\cdot|x-a_n|).$$

Поэтому каждое конечное множество натуральных чисел примитивно рекурсивно.

В п. 3.2 показано, что характеристические функции свойств «быть четным числом», «быть простым числом», «быть точным квадратом» примитивно рекурсивные. Поэтому примитивно рекурсивными являются и множества всех четных чисел, всех простых чисел, всех точных квадратов. Аналогичным путем легко убедиться, что примитивно рекурсивными будут совокупность степеней a^0, a^1, a^2, ... произвольного числа a и многие другие часто встречающиеся совокупности чисел.

Теорема 1. *Дополнение рекурсивного* (*примитивно рекурсивного*) *множества, а также объединение и пересечение любой конечной системы рекурсивных* (*примитивно рекурсивных*) *множеств суть множества рекурсивные* (*примитивно рекурсивные*).

В самом деле, пусть $f_1(x), \ldots, f_n(x)$ — характеристические функции множеств $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_n$. Тогда функции

$$f(x) = \overline{\mathrm{sg}}\, f_1(x),$$
$$g(x) = f_1(x)\cdot\ldots\cdot f_n(x),$$
$$h(x) = \mathrm{sg}\,(f_1(x) + \ldots + f_n(x))$$

будут характеристическими соответственно для дополнения множества $\boldsymbol{A}_1$, объединения и пересечения множеств $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_n$. Если $f_1(x), \ldots, f_n(x)$ частично рекурсивны или соответственно примитивно рекурсивны, то такими же будут и функции $f(x)$, $g(x)$, $h(x)$.

Теорема 2. *Если всюду определенная функция $f(x)$ частично рекурсивная (примитивно рекурсивная), то множество $\boldsymbol{A}$ решений уравнения*

$$f(x) = 0$$

рекурсивно (примитивно рекурсивно).

Действительно, характеристической функцией для множества $\boldsymbol{A}$ служит функция $\mathrm{sg}\, f(x)$, рекурсивная или примитивно рекурсивная вместе с функцией $f(x)$.

Совокупность всех значений, принимаемых некоторой примитивно рекурсивной функцией, в общем случае не будет ни примитивно рекурсивным, ни даже рекурсивным множеством. Эти совокупности будут изучены ниже под именем рекурсивно перечислимых множеств. Однако существует ряд простых признаков для того, чтобы совокупность значений примитивно рекурсивной (рекурсивной) функции была примитивно рекурсивной (рекурсивной). Один из них указывает

Теорема 3. *Если примитивно рекурсивная (общерекурсивная) функция $f(x)$ удовлетворяет условию*

$$f(x) \geqslant x \qquad (x = 0, 1, 2, \ldots),$$

в частности, если $f(x)$ монотонно возрастает, то совокупность $\boldsymbol{M}$ всех значений этой функции есть множество примитивно рекурсивное (рекурсивное).

В самом деле, из неравенства $f(y) \geqslant y$ следует, что функция

$$g(x) = \prod_{y=0}^{x} \mathrm{sg}\, |f(y) - x|$$

является характеристической для $\boldsymbol{M}$.

4.2. Рекурсивно перечислимые множества. Эти множества играют фундаментальную роль в теории рекурсивных функций. Существует несколько равносильных определений указанных множеств. В качестве исходного мы возьмем следующее определение.

Множество чисел $\boldsymbol{A}$ *называется рекурсивно перечислимым, если существует двуместная примитивно рекурсивная функция* $f(a, x)$ *такая, что уравнение* $f(a, x) = 0$ *имеет решение* x *тогда и только тогда, когда* $a \in \boldsymbol{A}$.

Далее (п. 6.1) будет показано, что в этом определении, на самом деле, в качестве функции $f(a, x)$ можно брать произвольную частично рекурсивную функцию. Однако этот результат будет получен нами лишь в результате длинной цепочки теорем. А пока мы, строго придерживаясь указанного определения, выведем из него ряд следствий.

Следствие 1. *Каждое примитивно рекурсивное множество рекурсивно перечислимо.*

Согласно определению характеристическая функция $f(x)$ произвольного примитивно рекурсивного множества $\boldsymbol{A}$ является примитивно рекурсивной. Но тогда примитивно рекурсивное уравнение

$$f(a) + x = 0$$

имеет решение x тогда и только тогда, когда $a \in \boldsymbol{A}$.

Следствие 2. *Пусть* $F(a, x_1, \ldots, x_n)$ — *примитивно рекурсивная функция от переменных* $a, x_1, \ldots, x_n$. *Множество тех значений параметра* a, *для которых уравнение*

$$F(a, x_1, \ldots, x_n) = 0 \qquad (1)$$

имеет хотя бы одно решение $x_1, \ldots, x_n$, *является рекурсивно перечислимым.*

Действительно, введем примитивно рекурсивную функцию

$$f(a, x) = F(a, c_{n1}(x), \ldots, c_{nn}(x)),$$

где $c_{ij}(x)$ — функции Кантора из п. 3.3. Так как

$$f(a, c(x_1, \ldots, x_n)) = F(a, x_1, \ldots, x_n),$$

то совокупность значений параметра a, для которых разрешимо уравнение $f(a, x) = 0$, совпадает с аналогичной совокупностью для уравнения (1) и потому эта последняя совокупность рекурсивно перечислима.

Теорема 1. *Непустое множество* $\boldsymbol{A}$ *тогда и только тогда рекурсивно перечислимо, когда оно совпадает с совокупностью всех значений некоторой примитивно рекурсивной функции.*

Пусть $f(x)$ — примитивно рекурсивная функция, $\boldsymbol{A}$ — множество всех ее значений. Левая часть уравнения

$$|f(x)-a|=0$$

примитивно рекурсивна и это уравнение имеет решение x тогда и только тогда, когда $a \in \boldsymbol{A}$. Следовательно, $\boldsymbol{A}$ рекурсивно перечислимо.

Обратно, пусть множество $\boldsymbol{A}$ непустое и рекурсивно перечислимое. Согласно определению это значит, что $\boldsymbol{A}$ есть совокупность тех значений параметра a, при которых разрешимо уравнение вида $F(a, x)=0$, где $F(a, x)$ — подходящая примитивно рекурсивная функция. Рассмотрим примитивно рекурсивную функцию

$$f(t)=\boldsymbol{l}(t)\,\overline{\mathrm{sg}}\,F(\boldsymbol{l}(t), \boldsymbol{r}(t))+b\cdot \mathrm{sg}\,F(\boldsymbol{l}(t), \boldsymbol{r}(t)), \qquad (2)$$

где $\boldsymbol{l}(t)$, $\boldsymbol{r}(t)$ — функции Кантора из п. 3.3, а b — какое-нибудь число из $\boldsymbol{A}$. Покажем, что $\boldsymbol{A}$ совпадает с совокупностью всех значений функции $f(t)$.

В самом деле, если при некотором t

$$F(\boldsymbol{l}(t), \boldsymbol{r}(t))=0, \qquad (3)$$

то $\boldsymbol{l}(t) \in \boldsymbol{A}$, а из (2) имеем

$$f(t)=\boldsymbol{l}(t) \in \boldsymbol{A}.$$

Если же (3) неверно, то из (2) получаем

$$f(t)=b \in \boldsymbol{A}.$$

Таким образом, все значения функции $f(t)$ входят в $\boldsymbol{A}$.

Пусть теперь $a \in \boldsymbol{A}$ и, значит, $F(a, x)=0$ для подходящего x. Полагая $t=\boldsymbol{c}(a, x)$ и принимая во внимание, что $\boldsymbol{l}(\boldsymbol{c}(a, x))=a$, получим из (2) $f(t)=a$. Следовательно, $\boldsymbol{A}$ совпадает с совокупностью значений функции $f(t)$ и теорема 1 доказана.

Указанное в этой теореме свойство рекурсивно перечислимых множеств иногда принимается за определение этих множеств.

Т е о р е м а 2. *Сумма и пересечение конечного числа рекурсивно перечислимых множеств являются рекурсивно перечислимыми множествами.*

Пусть A_i — множество тех значений параметра a, для которых существует решение x уравнения

$$f_i(a, x) = 0 \qquad (i = 1, \ldots, n),$$

где f_i — некоторая примитивно рекурсивная функция. Тогда совокупностью значений параметра a, при которых разрешимо относительно $x_1, \ldots, x_n$ уравнение

$$f_1(a, x_1) \cdot \ldots \cdot f_n(a, x_n) = 0 \tag{4}$$

или соответственно уравнение

$$f_1(a, x_1) + \ldots + f_n(a, x_n) = 0, \tag{5}$$

будет объединение или соответственно пересечение множеств A_i. Так как левые части уравнений (4), (5) примитивно рекурсивны, то согласно следствию 2 объединение и пересечение множеств A_i рекурсивно перечислимы.

По аналогии со свойствами рекурсивных множеств можно было бы предположить, что дополнение рекурсивно перечислимого множества есть рекурсивно перечислимое множество. Однако это неверно. В п. 6.3 будут построены рекурсивно перечислимые множества, дополнения которых не являются рекурсивно перечислимыми. Более того, именно этот случай следует считать типичным, так как справедлива

Теорема 3. (Пост). *Если какое-нибудь множество A рекурсивно перечислимо и его дополнение A' также рекурсивно перечислимо, то множества A и A' рекурсивны.*

Пусть множества A и A' являются совокупностями тех значений параметра a, для которых разрешимы уравнение $f(a, x) = 0$ и соответственно уравнение $g(a, x) = 0$, где f, g — примитивно рекурсивные функции. Так как для любого a одно из указанных уравнений заведомо разрешимо, то функция

$$h(a) = \mu_x(f(a, x)\, g(a, x) = 0) \tag{6}$$

всюду определенная и, следовательно, общерекурсивная. Поэтому функция

$$F(a) = \operatorname{sg} f(a, h(a))$$

также общерекурсивная. Из формулы (6) видно, что $F(a)$ — характеристическая функция для A, и потому A и A' рекурсивны.

В дополнение к теореме 1 докажем еще, что *совокупность значений, принимаемых произвольной многоместной примитивно рекурсивной функцией* $F(x_1, \ldots, x_n)$, *является рекурсивно перечислимым множеством.*

Рассмотрим одноместную функцию

$$f(x) = F(c_{n1}(x), \ldots, c_{nn}(x)). \qquad (7)$$

Эта функция примитивно рекурсивна. В силу (7) и соотношения

$$f(c(x_1, \ldots, x_n)) = F(x_1, \ldots, x_n),$$

совокупность значений функции f совпадает с совокупностью всех значений функции F и потому эта совокупность рекурсивно перечислимая.

4.3. Порожденные множества. Напомним некоторые понятия из п. 1.3. Пусть $\boldsymbol{M}$ — произвольное множество каких-то элементов. Каждая (частичная) функция $F(x_1, \ldots, x_n)$, определенная на $\boldsymbol{M}$, со значениями, принадлежащими тому же множеству $\boldsymbol{M}$, называется (частичной) *операцией* на $\boldsymbol{M}$. Подмножество $\boldsymbol{A}$ множества $\boldsymbol{M}$ называется замкнутым относительно частичной операции F, если для каждых значений $x_1, \ldots, x_n$, взятых из области определенности F и принадлежащих $\boldsymbol{A}$, значение $F(x_1, \ldots, x_n)$ также принадлежит $\boldsymbol{A}$.

Предположим, что в множестве $\boldsymbol{M}$, снабженном некоторой системой операций F_i, выделена какая-то совокупность $\boldsymbol{V}$ его элементов. Совокупность $\boldsymbol{V}^g$, являющаяся пересечением всех подмножеств множества $\boldsymbol{M}$, замкнутых относительно операций F_i и содержащих $\boldsymbol{V}$, называется совокупностью, *порожденной системой* $\boldsymbol{V}$ при помощи операций F_i. В п. 1.3 было показано, что $\boldsymbol{V}^g$ состоит из тех и только тех элементов $\boldsymbol{M}$, которые являются значениями термов, записываемых при помощи символов заданных операций F_i и символов элементов системы $\boldsymbol{V}$.

Для дальнейшего будет важен случай, когда множество $\boldsymbol{M}$ есть совокупность всех натуральных чисел, а операции F_i являются примитивно рекурсивными функциями.

Теорема 1. *Множество натуральных чисел* $\boldsymbol{V}^g$, *порожденное рекурсивно перечислимой совокупностью* $\boldsymbol{V}$ *с помощью конечной системы примитивно рекурсивных*

функций $F_i(x_1, \ldots, x_{m_i})$ $(i = 1, \ldots, s)$, *является рекурсивно перечислимым.*

Обозначим через $v(x)$ примитивно рекурсивную функцию, множество значений которой совпадает с $\boldsymbol{V}$. Согласно сказанному порожденное множество $\boldsymbol{V}^g$ состоит из значений термов $\mathfrak{a}(a_1, \ldots, a_m)$, записываемых с помощью функциональных знаков F_i и произвольных чисел $a_1, \ldots, a_m$ из $\boldsymbol{V}$. Так как $\boldsymbol{V}$ состоит из значений функции $v(x)$, то $\boldsymbol{V}^g$ состоит из значений термов вида $\mathfrak{a}(v(b_1), \ldots, v(b_m))$, где $b_1, \ldots, b_m$ — произвольные натуральные числа. Термы последнего вида условимся называть v-термами.

Все v-термы мы теперь перенумеруем. Номером терма $v(b)$ назовем число 3^b. Далее нумеруем индуктивно, восходя от более коротких термов к более длинным. Пусть термы $\mathfrak{a}_1, \ldots, \mathfrak{a}_{m_i}$ уже получили номера $c_1, \ldots, c_{m_i}$. Тогда номером терма $F_i(\mathfrak{a}_1, \ldots, \mathfrak{a}_{m_i})$ будем называть число $2^i p_1^{c_1} \ldots p_{m_i}^{c_{m_i}}$, где символы $p_0, p_1, p_2, \ldots$ обозначают последовательные простые числа 2, 3, 5, ...

Самые короткие v-термы — это термы вида $v(b)$. Их номера, как сказано, равны степеням числа 3. Номера остальных v-термов четные. Таким образом, не каждое натуральное число есть номер какого-то терма, но если натуральное число есть номер какого-нибудь v-терма, то этот терм однозначно восстанавливается путем разложений чисел на простые множители.

Мы хотим теперь построить вспомогательную функцию $f(x)$ таким образом, чтобы $f(n)$ было значением v-терма с номером n, если n является номером терма. С этой целью полагаем по определению $f(0) = v(0)$ и далее

$$f(n+1) = \begin{cases} F_1(f(\mathrm{ex}(1, n+1)), \ldots, f(\mathrm{ex}(m_1, n+1))), \\ \qquad\qquad \text{если } \mathrm{ex}(0, n+1) = 1, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ F_s(f(\mathrm{ex}(1, n+1)), \ldots, f(\mathrm{ex}(m_s, n+1))), \\ \qquad\qquad \text{если } \mathrm{ex}(0, n+1) = s, \\ v(\mathrm{ex}(1, n+1)) \text{ для остальных } n. \end{cases} \tag{1}$$

Здесь ex $(i, n+1)$ — показатель высшей степени числа p_i, на которую делится $n+1$ (см. п. 3.2).

Согласно п. 3.1 указанное определение функции $f(x)$ можно представить в следующей форме:

$$f(0)=v(0),$$
$$f(n+1)=G(n, f(\mathrm{ex}(1, n+1)), \ldots, f(\mathrm{ex}(t, n+1))), \qquad (2)$$

где $t=\max\{m_1, \ldots, m_s\}$ и

$$G(n, x_1, \ldots, x_t)=v(\mathrm{ex}(1, n+1))\,\mathrm{sg}\prod_{i=1}^{s}|\mathrm{ex}(0, n+1)-i|+$$
$$+\sum_{i=1}^{s}F_i(x_1, \ldots, x_{m_i})\,\overline{\mathrm{sg}}\,|\mathrm{ex}(0, n+1)-i|.$$

Так как функции ex $(i, n+1)$ удовлетворяют соотношениям

$$\mathrm{ex}(i, n+1)\leqslant n,$$

то равенства (2) представляют собой схему возвратной рекурсии, описанную в п. 3.2. Исходные функции ex $(i, n+1)$ и $G(n, x_1, \ldots, x_t)$ примитивно рекурсивны. Поэтому и функция $f(x)$ также примитивно рекурсивна.

Покажем, что $f(n)$ *равно значению v-терма с номером n, если n — вообще номер какого-то v-терма, и что $f(n)$ равно значению подходящего v-терма для любого n.*

Для $n=0$, 1 это заведомо верно, так как из (1) видно, что $f(0)=f(1)=v(0)$.

Пусть доказываемое утверждение истинно для каждого n, меньшего некоторого числа $a+1$. Рассмотрим $f(a+1)$. Могут представиться лишь следующие три случая.

а) Число $a+1$ есть номер v-терма вида $F_i(\mathfrak{a}_1, \ldots, \mathfrak{a}_{m_i})$.

Согласно принятой нумерации термов имеем

$$a+1=2^i\cdot p_1^{a_1}\ldots p_{m_i}^{a_{m_i}},$$

где $a_1, \ldots, a_{m_i}$ — номера термов $\mathfrak{a}_1, \ldots, \mathfrak{a}_{m_i}$. Из (1) находим

$$f(a+1)=F_i(f(a_1), \ldots, f(a_{m_i})).$$

Так как $a_j \leqslant a$, то по предположению $f(a_j)$ равно значению терма $\mathfrak{a}_j$ и, следовательно,

$$f(a+1) = F_i(\mathfrak{a}_1, \ldots, \mathfrak{a}_{m_i}).$$

Иначе говоря, в рассматриваемом случае $f(a+1)$ равно значению v-терма с номером $a+1$.

б) Число $a+1$ есть номер терма вида $v(c)$ и потому $a+1 = 3^c$. Из (1) находим

$$f(a+1) = v(c),$$

то есть снова получаем, что $f(a+1)$ равно значению v-терма с номером $a+1$.

в) Число $a+1$ не есть номер никакого v-терма. Пусть

$$\mathrm{ex}(i, a+1) = a_i \qquad (i = 0, 1, 2, \ldots).$$

По (1) получаем, что либо для некоторого i $(1 \leqslant i \leqslant s)$

$$f(a+1) = F_i(f(a_1), \ldots, f(a_{m_i})), \tag{3}$$

либо же

$$f(a+1) = v(a_1). \tag{4}$$

Так как $a_j \leqslant a$, то $f(a_j)$ равно значению какого-то v-терма $\mathfrak{a}_j$ (не обязательно с номером a_j). Но тогда в случае (3) $f(a+1)$ равно значению v-терма $F_i(\mathfrak{a}_1, \ldots$ $\ldots, \mathfrak{a}_{m_i})$, а в случае (4) $f(a+1)$ равно значению v-терма $v(a_1)$.

Итак, все условия индукции выполнены и утверждение можно считать доказанным. Из него вытекает, что порожденное множество V^g совпадает с совокупностью всех значений функции $f(n)$. Так как эта функция примитивно рекурсивная, то множество V^g рекурсивно перечислимо, что и требовалось.

4.4. Множества n-ок натуральных чисел. Наряду с множествами натуральных чисел обычно приходится рассматривать множества пар, троек и вообще n-ок натуральных чисел. Поэтому понятия и результаты предшествующих разделов мы распространим теперь на совокупности n-ок чисел.

Фиксируем некоторое $n \geqslant 2$. Через N^n обозначим совокупность всех n-ок натуральных чисел. В п. 3.3 каждой

n-ке натуральных чисел $\langle x_1, \ldots, x_n \rangle$ был поставлен в соответствие ее канторовский номер $c(x_1, \ldots, x_n)$, причем было показано, что это соответствие между $\boldsymbol{N}^n$ и $\boldsymbol{N}$ взаимно однозначное. Ставя в соответствие каждому множеству $\boldsymbol{A}$ n-ок чисел множество $c(\boldsymbol{A})$ номеров n-ок из $\boldsymbol{A}$, мы распространяем канторовское соответствие между $\boldsymbol{N}^n$ и $\boldsymbol{N}$ до взаимно однозначного канторовского соответствия между множествами чисел и множествами n-ок чисел. Отсюда возникает естественное

Определение. *Множество $\boldsymbol{A}$ n-ок натуральных чисел называется примитивно рекурсивным, рекурсивным или рекурсивно перечислимым, если таковым является множество $c(\boldsymbol{A})$ номеров всех n-ок из $\boldsymbol{A}$.*

Так как объединению и пересечению множеств n-ок отвечают объединение и пересечение совокупностей номеров этих n-ок, то из теорем п. 4.1 и п. 4.2 непосредственно получаем

Следствие 1. *Объединение и пересечение конечного числа рекурсивно перечислимых (рекурсивных, примитивно рекурсивных) множеств n-ок натуральных чисел являются рекурсивно перечислимыми (рекурсивными, примитивно рекурсивными) множествами.*

Дополнение рекурсивного (примитивно рекурсивного) множества n-ок есть рекурсивное (примитивно рекурсивное) множество.

Если дополнение $\boldsymbol{A}'$ рекурсивно перечислимого множества n-ок $\boldsymbol{A}$ рекурсивно перечислимо, то $\boldsymbol{A}$ и $\boldsymbol{A}'$ — рекурсивные множества.

Характеристической функцией множества $\boldsymbol{A}$ n-ок натуральных чисел называется n-местная функция $f(x_1, \ldots, x_n)$, равная 0 для n-ок $\langle x_1, \ldots, x_n \rangle$, входящих в $\boldsymbol{A}$, и равная 1 для n-ок, не входящих в $\boldsymbol{A}$. Если $c(\boldsymbol{A})$ — множество номеров всех n-ок из $\boldsymbol{A}$ и $g(x)$ — характеристическая функция множества $c(\boldsymbol{A})$, то из соотношений

$$g(x) = f(c_{n1}(x), \ldots, c_{nn}(x)),$$

$$g(c(x_1, \ldots, x_n)) = f(x_1, \ldots, x_n)$$

(см. п. 3.3) и примитивной рекурсивности канторовских функций c, c_{ni} следует, что *множество n-ок тогда и только тогда рекурсивно (примитивно рекурсивно), когда частично*

рекурсивна (примитивно рекурсивна) его характеристическая функция.

Аналогичным образом доказывается и

Теорема 1. *Если всюду определенная функция $F(x_1, \ldots, x_n)$ частично рекурсивна (примитивно рекурсивна), то совокупность n-ок $\langle x_1, \ldots, x_n \rangle$, удовлетворяющих уравнению*

$$F(x_1, \ldots, x_n) = 0,$$

является рекурсивным (примитивно рекурсивным) множеством.

Если функция $F(x_1, \ldots, x_n, y_1, \ldots, y_m)$ примитивно рекурсивна, то совокупность $\boldsymbol{A}$ *тех n-ок $\langle a_1, \ldots, a_n \rangle$, для которых уравнение*

$$F(a_1, \ldots, a_n, y_1, \ldots, y_m) = 0 \qquad (1)$$

имеет хотя бы одно решение $\langle y_1, \ldots, y_m \rangle$, является рекурсивно перечислимым множеством.

Первое утверждение очевидно (см. п. 4.1). Для доказательства второго утверждения вводим функцию

$$f(a, y) = F(c_{n1}(a), \ldots, c_{nn}(a), c_{m1}(y), \ldots, c_{mm}(y)).$$

Эта функция примитивно рекурсивна. Уравнение (1) разрешимо при заданных $a_1, \ldots, a_n$ тогда и только тогда, когда разрешимо относительно y уравнение

$$f(a, y) = 0, \qquad (2)$$

где a есть номер n-ки $\langle a_1, \ldots, a_n \rangle$. Но совокупность значений параметра a, для которых разрешимо уравнение (2), является рекурсивно перечислимым множеством. Поэтому рекурсивно перечислима и совокупность $\boldsymbol{A}$.

Теорема 2. *Для того чтобы непустая совокупность n-ок была рекурсивно перечислимой, необходимо и достаточно, чтобы она была совокупностью всех n-ок вида*

$$\langle f_1(x), \ldots, f_n(x) \rangle \qquad (x = 0, 1, \ldots), \qquad (3)$$

где $f_1(x), \ldots, f_n(x)$ — подходящие примитивно рекурсивные функции.

В самом деле, если функции $f_1(x), \ldots, f_n(x)$ примитивно рекурсивны, то совокупность номеров всех n-ок

вида (3) совпадает с совокупностью всех значений примитивно рекурсивной функции

$$f(x) = c(f_1(x), \ldots, f_n(x))$$

и потому совокупность n-ок вида (3) рекурсивно перечислима.

Обратно, пусть $c(A)$ — множество номеров n-ок рекурсивно перечислимой непустой совокупности. Согласно теореме 1 из п. 4.2 $c(A)$ есть совокупность значений некоторой примитивно рекурсивной функции $f(x)$. Но в таком случае A состоит из n-ок вида

$$\langle c_{n1}(f(x)), \ldots, c_{nn}(f(x))\rangle,$$

что и требовалось.

Графиком функции $F(x_1, \ldots, x_n)$ называется совокупность $(n+1)$-ок вида $\langle x_1, \ldots, x_n, F(x_1, \ldots, x_n)\rangle$, где значение $F(x_1, \ldots, x_n)$ должно быть определенным. Отсюда следует, что если функция F всюду определенная, то характеристической функцией ее графика будет функция

$$\mathrm{sg}\,|x_{n+1} - F(x_1, \ldots, x_n)|$$

и, значит, *графики примитивно рекурсивных и общерекурсивных функций являются соответственно множествами примитивно рекурсивными и общерекурсивными.*

Легко построить пример не всюду определенной функции, график которой примитивно рекурсивен. Более того, возможно построить и всюду определенную не примитивно рекурсивную функцию, график которой примитивно рекурсивен. Это довольно тонкое построение будет выполнено в п. 6.4, а сейчас мы докажем лишь следующее довольно очевидное утверждение:

Т е о р е м а 3. *Если график M всюду определенной функции $F(x_1, \ldots, x_n)$ рекурсивно перечислим, то функция F общерекурсивна.*

Согласно теореме 2 совокупность M состоит из $(n+1)$-ок вида $\langle f_1(t), \ldots, f_n(t), f_{n+1}(t)\rangle$, где $f_1, \ldots, f_{n+1}$ — подходящие примитивно рекурсивные функции. Функция F всюду определенная. Поэтому для любых $x_1, \ldots, x_n$ уравнение

$$|x_1 - f_1(t)| + \ldots + |x_n - f_n(t)| = 0$$

имеет хотя бы одно решение t, причем для каждого решения t этого уравнения имеем $F(x_1, \ldots, x_n) = f_{n+1}(t)$. Следовательно,

$$F(x_1, \ldots, x_n) = f_{n+1}(\mu_t(|x_1 - f_1(t)| + \ldots + |x_n - f_n(t)| = 0))$$

и, значит, функция $F(x_1, \ldots, x_n)$ общерекурсивная.

В заключение мы хотим сформулировать для n-ок натуральных чисел еще теорему о порожденных множествах.

Согласно п. 1.2 m-местной операцией (частичной) на множестве всех n-ок N^n называется отображение, которое последовательностям n-ок $\langle \mathfrak{x}_1, \ldots, \mathfrak{x}_m \rangle$ ставит в соответствие определенные n-ки $\mathfrak{y} = F(\mathfrak{x}_1, \ldots, \mathfrak{x}_m)$. Если

$$\mathfrak{x}_i = \langle x_{i1}, \ldots, x_{in} \rangle \qquad (i = 1, \ldots, m),$$

то каждая координата y_i n-ки $\mathfrak{y} = \langle y_1, \ldots, y_n \rangle$ будет функцией от координат n-ок $\mathfrak{x}_1, \ldots, \mathfrak{x}_n$:

$$y_i = f_i(x_{11}, \ldots, x_{1n}, \ldots, x_{m1}, \ldots, x_{mn}) \quad (i = 1, \ldots, n). \qquad (4)$$

Обратно, задавая произвольно функции f_i, мы сможем по n-кам $\mathfrak{x}_1, \ldots, \mathfrak{x}_m$ найти n-ку $\mathfrak{y}$ и тем самым восстановить операцию F. Если операция F частичная, то координатные функции f_i также частичные и наоборот.

Помимо координатных функций f_i с операцией F мы свяжем еще одну функцию f, которая будет называться *представляющей функцией для* F. Определяется она следующим образом.

Пусть даны некоторые числа $a_1, \ldots, a_m$. Берем n-ки $\mathfrak{x}_1, \ldots, \mathfrak{x}_m$ номеров $a_1, \ldots, a_m$, то есть n-ки

$$\mathfrak{x}_i = \langle c_{n1}(a_i), \ldots, c_{nn}(a_i) \rangle,$$

и вычисляем номер a n-ки $F(\mathfrak{x}_1, \ldots, \mathfrak{x}_m)$. Далее по определению полагаем $f(a_1, \ldots, a_m) = a$. Обратно, если известна функция $f(a_1, \ldots, a_m)$, то очевидным путем восстанавливается и операция $F(\mathfrak{x}_1, \ldots, \mathfrak{x}_m)$, представляемая функцией $f(a_1, \ldots, a_m)$.

Определение. *Операция $F(\mathfrak{x}_1, \ldots, \mathfrak{x}_m)$, заданная на множестве N^n всех n-ок натуральных чисел, называется частично рекурсивной, общерекурсивной или примитивно рекурсивной, если таковой является функция $f(a_1, \ldots, a_m)$, представляющая операцию F.*

Сравнивая определения представляющей и координатных функций, непосредственно получаем

С л е д с т в и е. *Для того чтобы операция* $F(x_1, \ldots, x_m)$, *определенная на* N^n, *была частично рекурсивной, общерекурсивной или примитивно рекурсивной, необходимо и достаточно, чтобы таковыми были все координатные функции* (4) *операции* F.

Например, рассматривая целые комплексные числа $x_1 + ix_2$ как пары натуральных чисел $\langle x_1, x_2 \rangle$, видим, что операция сложения этих чисел является примитивно рекурсивной.

Аналогично, пусть заданы какие-либо примитивно рекурсивные функции

$$g(x_1, x_2, y_1, y_2),\ h_i(x_1, x_2, y_1, y_2) \qquad (i = 1, 2, 3, 4).$$

Тогда двучленная операция $F(\mathfrak{x}, \mathfrak{y})$, определенная на N^2 условиями

$$F(\langle x_1, x_2 \rangle, \langle y_1, y_2 \rangle) = \begin{cases} \langle h_1(x_1, \ldots, y_2),\ h_2(x_1, \ldots, y_2) \rangle, \\ \qquad\qquad \text{если } g(x_1, \ldots, y_2) = 0, \\ \langle h_3(x_1, \ldots, y_2),\ h_4(x_1, \ldots, y_2) \rangle \\ \qquad\qquad \text{для остальных } x_1, \ldots, y_2, \end{cases}$$

будет примитивно рекурсивной.

В самом деле, указанные условия означают, что координатные функции f_1, f_2 для операции F задаются таблицами

$$f_1(x_1, x_2, y_1, y_2) = \begin{cases} h_1(x_1, x_2, y_1, y_2), \\ \qquad\qquad \text{если } g(x_1, \ldots, y_2) = 0, \\ h_3(x_1, x_2, y_1, y_2) \\ \qquad\qquad \text{для остальных } x_1, \ldots, y_2, \end{cases}$$

$$f_2(x_1, x_2, y_1, y_2) = \begin{cases} h_2(x_1, x_2, y_1, y_2), \\ \qquad\qquad \text{если } g(x_1, x_2, y_1, y_2) = 0, \\ h_4(x_1, x_2, y_1, y_2) \\ \qquad\qquad \text{для остальных } x_1, x_2, y_1, y_2, \end{cases}$$

показывающими (см. п. 3.1), что эти координатные функции, а вместе с ними и операция F, являются примитивно рекурсивными.

Теорема (п. 4.3) о порожденных множествах чисел легко переносится без изменений и на множества n-ок чисел.

Теорема 4. *Пусть на множестве* $\boldsymbol{N}^n$ *всех* n-*ок натуральных чисел задано конечное число примитивно рекурсивных операций* $F_i(\mathfrak{x}_1, \ldots, \mathfrak{x}_{m_i})$ $(i = 1, \ldots, s)$ *и пусть задано некоторое рекурсивно перечислимое множество* $\boldsymbol{V}$ n-*ок из* $\boldsymbol{N}^n$. *Тогда совокупность* $\boldsymbol{V}^g$ n-*ок, порожденная элементами* $\boldsymbol{V}$ *с помощью операций* F_i, *является также рекурсивно перечислимой.*

Пусть $\boldsymbol{c}(\boldsymbol{V})$, $\boldsymbol{c}(\boldsymbol{V}^g)$ – совокупности канторовских номеров всех n-ок из $\boldsymbol{V}$ и соответственно из $\boldsymbol{V}^g$. Обозначим через $f_i(x_1, \ldots, x_{m_i})$ функцию, представляющую операцию $F_i(\mathfrak{x}_1, \ldots, \mathfrak{x}_{m_i})$ $(i = 1, \ldots, s)$. Остается лишь убедиться, что множество чисел $\boldsymbol{c}(\boldsymbol{V}^g)$ порождается множеством $\boldsymbol{c}(\boldsymbol{V})$ с помощью операций $f_1, \ldots, f_s$, так как тогда ввиду теоремы 1 из п. 4.3 $\boldsymbol{c}(\boldsymbol{V}^g)$ и $\boldsymbol{V}^g$ будут рекурсивно перечислимыми.

Множество $\boldsymbol{V}^g$ есть совокупность значений термов, записываемых при помощи символов операций F_i и обозначений n-ок из $\boldsymbol{V}$. Пусть $\mathfrak{a}(\mathfrak{x}_1, \ldots, \mathfrak{x}_n)$ — один из таких термов. Заменяя в нем символы операций F_i символами представляющих функций f_i, а символы n-ок $\mathfrak{x}_i$ символами их номеров, получим терм, значение которого будет равно номеру значения терма $\mathfrak{a}$. Таким образом, совокупность $\boldsymbol{c}(\boldsymbol{V}^g)$ является множеством значений термов, записываемых при помощи символов функций f_i и символов чисел из $\boldsymbol{c}(\boldsymbol{V})$, то есть $\boldsymbol{c}(\boldsymbol{V}^g)$ порождается совокупностью $\boldsymbol{c}(\boldsymbol{V})$ при помощи функций f_i, что и требовалось.

Согласно п. 1.3 множество $\boldsymbol{M}$, снабженное конечной системой определенных на нем операций $F_i(x_1, \ldots, x_{m_i})$, называется алгеброй, символически обозначаемой через $\mathfrak{M} = \langle \boldsymbol{M}; F_1, \ldots, F_s \rangle$.

Если основное множество $\boldsymbol{M}$ алгебры $\mathfrak{M}$ является совокупностью всех n-ок натуральных чисел, а основные операции F_i являются примитивно рекурсивными, то алгебра $\mathfrak{M}$ называется примитивно рекурсивной. Теорема 4 может быть теперь переформулирована следующим образом: каждое рекурсивно перечислимое множество элементов примитивно рекурсивной алгебры порождает в этой алгебре рекурсивно перечислимую подалгебру.

Примеры и задачи

1. Каждое бесконечное рекурсивно перечислимое множество содержит бесконечное рекурсивное подмножество.

2. *Проекции.* Геометрически n-ки чисел истолковывают обычно как точки n-мерного пространства; числа $x_1, \ldots, x_n$ называют координатами n-ки $\langle x_1, \ldots, x_n \rangle$. Пусть $1 \leqslant i_1 < \ldots < i_m \leqslant n$. m-ку $\langle x_{i_1}, \ldots, x_{i_m} \rangle$ называют проекцией точки $\langle x_1, \ldots, x_n \rangle$ на координатную $\langle i_1, \ldots, i_m \rangle$-плоскость. Совокупность проекций всех точек множества A на $\langle i_1, \ldots, i_m \rangle$-плоскость называется проекцией множества A на эту плоскость.

Показать, что проекция рекурсивно перечислимого множества есть множество рекурсивно перечислимое и что каждое рекурсивно перечислимое n-мерное множество есть проекция подходящего $(n+1)$-мерного примитивно рекурсивного множества.

3. Вещественное число α, имеющее вид

$$\alpha = a_0 + a_1 \cdot 10^{-1} + a_2 \cdot 10^{-2} + \ldots,$$

где $a_0, a_1, \ldots$ целые, $0 \leqslant a_i \leqslant 9$ для $i = 1, 2, \ldots$, называется рекурсивным (примитивно рекурсивным), если рекурсивна (примитивно рекурсивна) функция $f_\alpha(x)$, определенная условиями

$$f_\alpha(0) = |a_0|, \quad f_\alpha(n+1) = a_{n+1}.$$

Показать, что все алгебраические вещественные числа примитивно рекурсивны и что все рекурсивные вещественные числа образуют алгебраически замкнутое поле.

ГЛАВА III

ОБЩЕРЕКУРСИВНЫЕ И ЧАСТИЧНО РЕКУРСИВНЫЕ ФУНКЦИИ

Центральное понятие теории алгоритмов — это понятие функции, значения которой вычислимы с помощью алгоритма. Согласно тезису Чёрча (п. 2.3) алгоритмически вычислимые функции отождествляются с частично рекурсивными функциями. Настоящая глава посвящена доказательствам фундаментальных теорем о рекурсивных и частично рекурсивных функциях. Теоремы эти образуют основу теории рекурсивных функций.

§ 5. ОБЩЕРЕКУРСИВНЫЕ ФУНКЦИИ

До сих пор не было установлено, что класс общерекурсивных функций действительно шире класса примитивно рекурсивных функций. В настоящем параграфе этот пробел восполняется фактическим построением общерекурсивной функции, не являющейся примитивно рекурсивной. Более того, построенная общерекурсивная функция от двух переменных оказывается универсальной для всех одноместных примитивно рекурсивных функций и в дальнейшем изложении играет особую роль. Конструирование универсальной функции проходит весьма просто при помощи одного специального вида рекурсии, с изучения свойств которого мы и начнем изложение материала этого параграфа.

5.1. Рекурсии 2-й ступени. Числа натурального чяда 0, 1, 2, . . . образуют естественным образом упорядоченную последовательность. Эта упорядоченность играет фундаментальную роль в задании функций с помощью

примитивной рекурсии. Именно, если определяемая функция многоместная, то из аргументов выбирается один, задается значение функции, когда этот аргумент равен 0, и дается правило, как найти значение функции в точке $n+1$, если известны ее значения при меньших значениях аргумента. Допустим теперь, что мы хотим аналогичным способом задать функцию двух аргументов $F(x, y)$, но хотим вести рекурсию не по одному аргументу, а сразу по обоим. Тогда мы предварительно должны как-то упорядочить все пары чисел $\langle x, y\rangle$. Конечно, упорядочивать их можно многими способами. Остановимся на словарном упорядочении: пара $\langle x, y\rangle$ будет считаться предшествующей паре $\langle x_1, y_1\rangle$, если либо $x < x_1$, либо $x = x_1$, а $y < y_1$. Выписывая пары в только что установленном порядке, получим

$$\langle 0, 0\rangle < \langle 0, 1\rangle < \ldots < \langle 1, 0\rangle < \langle 1, 1\rangle < \ldots$$
$$\ldots < \langle 2, 0\rangle < \langle 2, 1\rangle < \ldots \qquad (1)$$

В этом расположении пары вида $\langle n, 0\rangle$ играют особую роль — у них нет непосредственно предшествующей пары, а пара $\langle 0, 0\rangle$ вообще является начальной.

Вернемся теперь к функции $F(x, y)$, которую мы хотим задать рекурсией сразу по двум аргументам. Принимая во внимание расположение (1), мы должны сначала задать $F(0, 0)$, положив его равным некоторому числу a. Затем мы должны как-то выразить $F(x, y)$ через значения этой же функции для предыдущих пар значений аргументов. Здесь возможны следующие случаи:

а) Требуется найти $F(0, x)$. Предыдущие значения функции имеют вид $F(0, y)$, где $y < x$.

б) Требуется найти $F(n+1, 0)$. Предыдущие значения функции имеют вид $F(n_1, x)$, где $n_1 \leqslant n$, а x произвольно. В частности, если «предыдущие» значения функции F известны, то можно считать известными и значения выражений вида

$$F(f_1(n), g(x)); \qquad F(f_1(n), F(f_2(n), g(x)));$$
$$F(f_1(n), F(f_2(n), F(f_3(n), x))); \ldots, \qquad (2)$$

где $f_i(n)$, $g(x)$ — какие-либо наперед заданные функции,

удовлетворяющие условиям

$$f_i(n) \leqslant n \qquad (i = 1, 2, \ldots). \tag{3}$$

в) Требуется найти $F(n+1,\ x+1)$. Предыдущие значения функции F здесь могут быть следующих типов:

$$F(n+1,\ f_1(x));\ F(f_1(n),\ F(f_2(n),\ g(x)));$$
$$F(f_1(n),\ F(n+1,\ f_2(x)));\ \ldots, \tag{4}$$

где f_i — наперед заданные функции, подчиненные условиям (3).

Таким образом, чтобы задать функцию $F(x, y)$ рекурсией по паре переменных, мы можем задать выражения для $F(n+1,\ 0)$, $F(n+1,\ x+1)$ через члены вида (2), (4), а также задать функцию одной переменной $F(0,\ x+1)$ и число $F(0,\ 0)$. В полном объеме этот способ задания нам далее не потребуется. В частности, нам не будут нужны члены вида $F(f, (n),\ F(f_2(n),\ F(f_3(n),\ x)))$ и аналогичной более сложной структуры. Мы ограничимся рассмотрением лишь следующего способа, который только и встретится нам далее. Этот способ мы будем называть рекурсией 2-й ступени.

Определение. *Пусть заданы всюду определенные функции*

$$G(x_0,\ x_1,\ \ldots,\ x_m),\quad H(x_0,\ x_1,\ \ldots,\ x_k),\ h(x)$$

и всюду определенные функции $f_i(x)$, $g_j(x)$, *удовлетворяющие требованиям*

$$f_i(x) \leqslant x,\ g_j(x) \leqslant x\ \ (i = 1, \ldots, m;\ j = 1, \ldots, k). \tag{5}$$

Говорят, что всюду определенная функция $F(x, y)$ *возникает из функций* G, H, h, f_i, g_j *рекурсией 2-й ступени, если*

$$\begin{gathered} F(n+1,\ x+1) = \\ = G(n+1, x, F(f_1(n), F(n+1, x)), F(f_2(n), F(f_3(n), x+1)), \\ F(f_4(n),\ x+1), \ldots, F(f_m(n),\ x+1)), \\ F(n+1,\ 0) = \\ = H(n+1, F(g_1(n), F(g_2(n), 0)), F(n, 0), F(g_3(n), 0), \ldots \\ \ldots, F(g_k(n), 0)), \\ F(0,\ x) = h(x) \end{gathered} \tag{6}$$

для всех значений n, x.

Например, в п. 5.3 мы детально изучим функцию $B(x, y)$, определенную схемой

$$B(n+1,\ x+1) = B(n,\ B(n+1,\ x)),$$
$$B(n+1,\ 0) = \operatorname{sg} n,$$
$$B(0,\ x) = 2 + x.$$

Сравнение этой схемы со схемой (6) показывает, что $B(x, y)$ возникает рекурсией 2-й ступени из функций $G(n, x, y) = y$,

$$f_1(n) = n, \quad h(x) = 2 + x, \quad H(n) = \operatorname{sg} n.$$

Покажем теперь, что каковы бы ни были функции G, H, h и функции f_i, g_i, удовлетворяющие условиям (5), существует функция $F(x, y)$ и только одна, удовлетворяющая схеме (6).

Построение функции $F(x, y)$ выполняется очевидным путем. По определению для всех x полагаем $F(0, x) = h(x)$. Далее, если для всех пар $\langle u, v\rangle$, меньших пары $\langle n+1, 0\rangle$, значения $F(u, v)$ определены и для них условия (6) соблюдены, то значение $F(n+1, 0)$ определяем второй формулой из (6). Наконец, если значения F определены для всех пар, меньших пары $\langle n+1,\ x+1\rangle$, то значение $F(n+1,\ x+1)$ определяем первой формулой из (6). Ясно, что полученная таким образом функция удовлетворяет условиям (6). Поскольку выбиравшиеся значения для $F(n, x)$ однозначно вытекали из формул (6), то формулы (6) определяют F однозначно.

Для дальнейшего принципиальное значение имеет

Теорема 1 (о рекурсии 2-й ступени). *Если указанные в определении рекурсии 2-й ступени функции G, H, h, f_i, g_j примитивно рекурсивны, то возникающая из них посредством рекурсии* (6) *функция F является общерекурсивной.*

Чтобы сократить запись формул, мы докажем эту теорему для $m = 4$, $k = 3$. В общем случае рассуждения остаются теми же.

Обозначим через $\boldsymbol{M}$ график функции F, то есть совокупность троек чисел вида $\langle x, y, F(x, y)\rangle$. Согласно п. 4.4 теорема 1 будет доказана, если мы обнаружим, что совокупность $\boldsymbol{M}$ рекурсивно перечислимая. Посмотрим, что означают равенства (6) для $\boldsymbol{M}$.

Первое из равенств (6), очевидно, можно пересказать так: *если тройки*

$$\langle n+1,\ x,\ u\rangle,\ \langle f_1(n),\ u,\ v\rangle,\ \langle f_3(n),\ x+1,\ p\rangle,\\ \langle f_2(n),\ p,\ q\rangle,\ \langle f_4(n),\ x+1,\ w\rangle \tag{7}$$

принадлежат $\boldsymbol{M}$, *то*

$$\langle n+1,\ x+1,\ G(n+1,\ x,\ v,\ q,\ w)\rangle \in \boldsymbol{M}. \tag{8}$$

Второе из равенств (6) равносильно утверждению: *если тройки*

$$\langle g_2(n),\ 0,\ u\rangle,\ \langle g_1(n),\ u,\ v\rangle,\ \langle n,\ 0,\ w\rangle,\ \langle g_3(n),\ 0,\ p\rangle \tag{9}$$

принадлежат $\boldsymbol{M}$, *то*

$$\langle n+1,\ 0,\ H(n+1,\ v,\ w,\ p)\rangle \in \boldsymbol{M}. \tag{10}$$

Наконец, последнее из равенств (6) утверждает, что $\boldsymbol{M}$ *содержит все тройки вида* $\langle 0,\ x,\ h(x)\rangle$.

Обозначим через $\boldsymbol{N}^3$ совокупность всех троек чисел. Мы хотим определить на $\boldsymbol{N}^3$ такие операции $\mathscr{A}(u_1,\ u_2,\ u_3,\ u_4,\ u_5)$, $\mathscr{B}(u_1,\ u_2,\ u_3,\ u_4)$, которые для наборов троек (7), (9) имели бы своими значениями соответственно тройки (8) и (10). С этой целью полагаем по определению для произвольных x_i, y_i, z_i

$$\mathscr{A}(\langle x_1,\ y_1,\ z_1\rangle,\ \ldots,\ \langle x_5,\ y_5,\ z_5\rangle) =\\ = \langle x_1,\ y_3,\ G(x_1,\ y_1,\ z_2,\ z_4,\ z_5)\rangle, \tag{11}$$

если выполнены условия

$$\left.\begin{array}{llll} x_2 = f_1(x_1 \dot{-} 1), & y_2 = z_1, & x_3 = f_3(x_1 \dot{-} 1), & y_3 = y_1+1,\\ x_4 = f_2(x_1 \dot{-} 1), & y_4 = z_3, & x_5 = f_4(x_1 \dot{-} 1), & y_5 = y_1+1, \end{array}\right. \tag{12}$$

и полагаем

$$\mathscr{A}(\langle x_1,\ y_1,\ z_1\rangle,\ \ldots,\ \langle x_5,\ y_5,\ z_5\rangle) = \langle 0,\ 0,\ h(0)\rangle, \tag{13}$$

если условия (12) не выполнены.

Аналогично полагаем по определению

$$\mathscr{B}(\langle x_1, y_1, z_1\rangle, \ldots, \langle x_4, y_4, z_4\rangle) = \langle x_3+1, 0, H(x_3+1, z_2, z_3, z_4)\rangle, \tag{14}$$

если выполнены условия

$$x_1 = g_2(x_3),\quad x_2 = g_1(x_3),\quad y_2 = z_1,\quad x_4 = g_3(x_3), \tag{15}$$

и полагаем

$$\mathscr{B}(\langle x_1, y_1, z_1\rangle, \ldots, \langle x_4, y_4, z_4\rangle) = \langle 0, 0, h(0)\rangle, \quad (16)$$

если условия (15) для $x_1, y_1, \ldots, z_4$ не удовлетворяются.

Так как функции G, H, h, f_i, g_i примитивно рекурсивны, то согласно п. 4.4 операции $\mathscr{A}$ и $\mathscr{B}$ также примитивно рекурсивны.

Обозначим через $\boldsymbol{M}_0$ совокупность троек вида $\langle 0, x, h(x)\rangle$. Из примитивной рекурсивности функции $h(x)$ следует, что совокупность $\boldsymbol{M}_0$ рекурсивно перечислима.

Согласно теореме о порождаемых совокупностях (п. 4.4) совокупность $\boldsymbol{M}^*$ всех троек, которые можно получить из троек множества $\boldsymbol{M}_0$ с помощью операций $\mathscr{A}$, $\mathscr{B}$, является рекурсивно перечислимой и нам остается доказать лишь совпадение совокупности $\boldsymbol{M}^*$ с графиком $\boldsymbol{M}$.

Прежде всего замечаем, что $\boldsymbol{M}$ содержит порождающее множество $\boldsymbol{M}_0$. Кроме того, условия (11), (13) и (14), (16) показывают, что совокупность $\boldsymbol{M}$ замкнута относительно операций $\mathscr{A}$, $\mathscr{B}$. Совокупность $\boldsymbol{M}^*$ по определению есть наименьшая совокупность, замкнутая относительно $\mathscr{A}$, $\mathscr{B}$ и содержащая $\boldsymbol{M}_0$. Таким образом, $\boldsymbol{M}^* \subseteq \boldsymbol{M}$.

Обратное включение $\boldsymbol{M} \subseteq \boldsymbol{M}^*$ проще всего доказать индукцией по параметрам n, x троек $\langle n, x, F(n, x)\rangle$, составляющих график $\boldsymbol{M}$. Для $n = 0$ график $\boldsymbol{M}$ содержит лишь тройки $\langle 0, x, h(x)\rangle$. Все они входят в $\boldsymbol{M}^*$. Пусть для некоторого n все тройки $\langle r, x, F(r, x)\rangle$, удовлетворяющие условию $r \leqslant n$, входят в $\boldsymbol{M}^*$. Покажем, что тогда и тройка $\langle n+1, 0, F(n+1, 0)\rangle$ входит в $\boldsymbol{M}^*$. Полагая

$$u = F(g_2(n), 0), \quad v = F(g_1(n), u),$$
$$w = F(n, 0), \qquad p = F(g_3(n), 0),$$

имеем в силу (6)

$$F(n+1, 0) = H(n+1, v, w, p).$$

Так как $g_i(n) \leqslant n$, то все тройки (9) по предположению входят в $\boldsymbol{M}^*$. Применяя к тройкам (9) операцию $\mathscr{B}$, получим тройку $\langle n+1, 0, F(n+1, 0)\rangle$. Совокупность $\boldsymbol{M}^*$ замкнута относительно операции $\mathscr{B}$. Следовательно, тройка $\langle n+1, 0, F(n+1, 0)\rangle$ принадлежит $\boldsymbol{M}^*$.

Аналогично доказывается и утверждение, что если для данных n, x все тройки $\langle r, s, F(r, s)\rangle$ с условием $\langle r, s\rangle < \langle n+1, x+1\rangle$ содержатся в M^*, то и тройка $\langle n+1, x+1, F(n+1, x+1)\rangle$ содержится в M^*. Тем самым включение $M \subseteq M^*$, а вместе с ним и равенство $M = M^*$ доказаны.

5.2. Универсальная общерекурсивная функция. Рассмотрим произвольную систему $\mathfrak{S}$ частичных n-местных функций. Частичная функция $F(x_0, x_1, \ldots, x_n)$ от $n+1$ переменных называется *универсальной* для семейства $\mathfrak{S}$, если выполнены следующие два условия:

а) *для каждого фиксированного числа i n-местная функция $F(i, x_1, \ldots, x_n)$ принадлежит $\mathfrak{S}$;*

б) *для каждой функции $f(x_1, \ldots, x_n)$ из $\mathfrak{S}$ существует такое число i, что для всех $x_1, \ldots, x_n$*

$$F(i, x_1, \ldots, x_n) = f(x_1, \ldots, x_n).$$

Иначе говоря, функция $F(x_0, x_1, \ldots, x_n)$ универсальна для семейства $\mathfrak{S}$, если все функции из $\mathfrak{S}$ можно расположить в последовательность

$$F(0, x_1, \ldots, x_n), F(1, x_1, \ldots, x_n), \ldots, F(i, x_1, \ldots, x_n), \ldots \tag{1}$$

Число i называется *номером* функции $F(i, x_1, \ldots, x_n)$, а полученная таким образом нумерация функций семейства $\mathfrak{S}$ называется нумерацией, отвечающей универсальной функции F. Обратно, если задана какая-либо нумерация системы $\mathfrak{S}$, то есть если задано какое-нибудь отображение $i \to f_i$ натурального ряда на $\mathfrak{S}$, то функция $F(x_0, x_1, \ldots, x_n)$, определенная формулой

$$F(x, x_1, \ldots, x_n) = f_x(x_1, \ldots, x_n),$$

является универсальной для $\mathfrak{S}$.

Заметим, что рассматриваемые здесь нумерации, в отличие от нумераций пар, троек и т. д. из п. 3.3, не одно-однозначные: различные натуральные числа j, i могут быть номерами одной и той же функции из $\mathfrak{S}$.

Существование достаточно «хорошей» универсальной функции является важной характеристикой системы $\mathfrak{S}$. Например, если система $\mathfrak{S}$ состоит только из всюду

определенных функций, то существует очень простой метод (метод диагоналей Кантора) построения всюду определенной функции, не входящей в $\mathfrak{S}$.

В самом деле, пусть $F(x, x_1, \ldots, x_n)$ — универсальная функция для семейства $\mathfrak{S}$ некоторых всюду определенных функций от n переменных. Из определения следует, что эта универсальная функция будет также всюду определенной. Вводим новую функцию g, полагая

$$g(x_1, \ldots, x_n) = F(x_1, x_1, x_2, \ldots, x_n) + 1. \qquad (2)$$

Функция $g(x_1, \ldots, x_n)$ не входит в $\mathfrak{S}$.

Действительно, если бы g принадлежала семейству $\mathfrak{S}$, то при подходящем натуральном i для всех $x_1, \ldots, x_n$ было бы истинно равенство

$$F(i, x_1, \ldots, x_n) = F(x_1, x_1, x_2, \ldots, x_n) + 1.$$

Однако при $x_1 = i$ оно обращается в противоречивое соотношение

$$F(i, i, \ldots, x_n) = F(i, i, \ldots, x_n) + 1,$$

что и доказывает наше утверждение.

Отметим два важных следствия этого утверждения. Пусть $\mathfrak{S}$ — система всех n-местных рекурсивных или соответственно примитивно рекурсивных функций. Равенство (2) показывает, что если функция $F(x_0, x_1, \ldots, x_n)$ рекурсивна или примитивно рекурсивна, то таковой является и функция $g(x_1, \ldots, x_n)$. Но эта функция не входит в $\mathfrak{S}$ и, значит, она не рекурсивна или соответственно не примитивно рекурсивна. Тем самым пришли к следующей простой, но существенной теореме.

Теорема 1. *Система всех n-местных общерекурсивных функций не имеет общерекурсивной универсальной функции. Система всех n-местных примитивно рекурсивных функций не имеет примитивно рекурсивной универсальной функции ($n = 1, 2, \ldots$).*

Нижеследующая теорема принадлежит к числу основных теорем теории рекурсивных функций.

Теорема 2. *Система всех одноместных примитивно рекурсивных функций имеет общерекурсивную универсальную функцию.*

Чтобы построить искомую универсальную функцию, нужно все одноместные примитивно рекурсивные функции расположить в последовательность вида (1). Для этой цели мы воспользуемся теоремой Робинсона из п. 3.5, согласно которой все одноместные примитивно рекурсивные функции можно получить из функций $\boldsymbol{s}(x) = x + 1$ и $\boldsymbol{q}(x) = x - [\sqrt{x}]^2$ операциями сложения, суперпозиции и итерирования функций. Эти операции были обозначены в п. 3.5 символами $+$, $*$, $\boldsymbol{J}$. Упомянутая теорема Робинсона утверждает, что каждая одноместная примитивно рекурсивная функция является значением подходящего терма, составленного из индивидуальных символов $\boldsymbol{s}$, $\boldsymbol{q}$ и функциональных символов. Обратное также верно: значение каждого терма упомянутого вида есть примитивно рекурсивная одноместная функция. Поэтому, нумеруя термы, мы одновременно занумеруем и все примитивно рекурсивные одноместные функции. Номер терма $\mathfrak{a}$ условимся обозначать через $N(\mathfrak{a})$.

Для самых коротких термов $\boldsymbol{s}$ и $\boldsymbol{q}$ полагаем по определению

$$N(\boldsymbol{s}) = 1, \quad N(\boldsymbol{q}) = 3. \tag{3}$$

Далее определяем номера термов индуктивно, восходя от коротких термов к более длинным. А именно, если для каких-нибудь термов $\mathfrak{a}$, $\mathfrak{b}$ мы уже знаем их номера $N(\mathfrak{a}) = a$, $N(\mathfrak{b}) = b$, то по определению полагаем

$$\begin{aligned} N(\mathfrak{a} + \mathfrak{b}) &= 2 \cdot 3^a \cdot 5^b, \\ N(\mathfrak{a} * \mathfrak{b}) &= 4 \cdot 3^a \cdot 5^b, \\ N(\boldsymbol{J}\mathfrak{a}) &= 8 \cdot 3^a. \end{aligned} \tag{4}$$

Например, термы $\boldsymbol{s} + \boldsymbol{J}\boldsymbol{q}$ и $\boldsymbol{J}(\boldsymbol{s} + \boldsymbol{s})$ будут иметь номера $2 \cdot 3 \cdot 5^{8 \cdot 3^3}$ и соответственно $8 \cdot 3^{30}$.

Можно считать, следовательно, что мы определили для каждого терма $\mathfrak{a}$ его номер. Ясно, что далеко не каждое натуральное число является номером некоторого терма, но если число a есть номер терма, то только одного, и этот терм легко восстанавливается путем разложения a на простые множители.

Определяем теперь новую функцию $F(n, x)$ формулой

$$F(n, x) = f_n(x), \tag{5}$$

где f_n — терм с номером n. Так как не все натуральные числа являются номерами термов, то функция $F(n, x)$ также определена не для всех значений n и потому является частичной. Однако для тех значений n, для которых F определена, из способа нумерации термов (3) и (4) вытекают следующие соотношения:

$$F(n, x) = \begin{cases} f_a(x) + f_b(x), & \text{если} \quad n = 2\cdot 3^a \cdot 5^b; \\ f_a(f_b(x)), & \text{если} \quad n = 4\cdot 3^a \cdot 5^b; \\ f_a(f_n(x-1)), & \text{если} \quad n = 8\cdot 3^a,\ x > 0; \\ 0, & \text{если} \quad n = 8\cdot 3^a,\ x = 0; \\ q(x), & \text{если} \quad n = 3; \\ s(x), & \text{если} \quad n = 1. \end{cases} \tag{6}$$

Принимая во внимание равенство (5), мы можем соотношения (6) переписать в виде

$$F(n, x) = \\ = \begin{cases} F(\mathrm{ex}_1 n, x) + F(\mathrm{ex}_2 n, x), & \text{если} \quad \mathrm{ex}_0 n = 1; \\ F(\mathrm{ex}_1 n, F(\mathrm{ex}_2 n, x)), & \text{если} \quad \mathrm{ex}_0 n = 2; \\ F(\mathrm{ex}_1 n, F(n, x-1)), & \text{если} \quad \mathrm{ex}_0 n = 3,\ x > 0; \\ 0, & \text{если} \quad \mathrm{ex}_0 n = 3,\ x = 0; \\ Q(n, x) & \text{для остальных } n, x, \end{cases} \tag{7}$$

где для краткости положено

$$Q(n, x) = s(x)\,\overline{\mathrm{sg}}\,|n-1| + q(x)\,\overline{\mathrm{sg}}\,|n-3|.$$

Подчеркнем еще раз, что соотношения (7) установлены лишь для тех значений n, которые являются номерами термов. Мы хотим теперь найти всюду определенную функцию $D(n, x)$, удовлетворяющую тем же соотношениям (7) для всех значений n, x. Возникают вопросы: существует ли такая функция D и совпадают ли значения $D(n, x)$ со значениями $F(n, x)$ в области определенности F, если функция D существует? Чтобы ответить на первый

вопрос, мы перепишем условия (7) для функции D в более подробной форме:

$$D(n+1, x+1) = \begin{cases} D(f_1(n), x+1) + D(f_2(n), x+1), & \text{если } \mathrm{ex}_0(n+1) = 1; \\ D(f_1(n), D(f_2(n), x+1)), & \text{если } \mathrm{ex}_0(n+1) = 2; \\ D(f_1(n), D(n+1, x)), & \text{если } \mathrm{ex}_0(n+1) = 3; \\ Q(n+1, x+1) & \text{для остальных } n, \end{cases} \tag{8}$$

$$D(n+1, 0) = \begin{cases} D(f_1(n), 0) + D(f_2(n), 0), & \text{если } \mathrm{ex}_0(n+1) = 1; \\ D(f_1(n), D(f_2(n), 0)), & \text{если } \mathrm{ex}_0(n+1) = 2; \\ 0 & \text{для остальных } n, \end{cases} \tag{9}$$

$$D(0, x) = 0, \tag{10}$$

где положено

$$f_1(n) = \mathrm{ex}_1(n+1), \quad f_2(n) = \mathrm{ex}_2(n+1). \tag{11}$$

С помощью приема, рассмотренного в п. 3.1, мы можем схемы (8), (9) переписать в виде простых равенств и таким образом заменить условия (8), (9), (10) условиями

$$\begin{aligned} D(n+1, x+1) &= G(n+1, x, D(f_1(n), D(n+1, x)), D(f_1(n), \\ &D(f_2(n), x+1)), D(f_1(n), x+1), D(f_2(n), x+1)), \\ D(n+1, 0) &= H(n+1, D(f_1(n), D(f_2(n), 0)), D(f_1(n), 0), \\ &D(f_2(n), 0)), \end{aligned} \tag{12}$$

где

$$\begin{aligned} G(m, x, y, z, u, v) &= (u+v)\,\overline{\mathrm{sg}}\,|\mathrm{ex}_0 m - 1| + z\,\overline{\mathrm{sg}}\,|\mathrm{ex}_0 m - 2| + \\ &+ y\,\overline{\mathrm{sg}}\,|\mathrm{ex}_0 m - 3| + Q(m, x)\,\mathrm{sg}(|\mathrm{ex}_0 m - 1|\,|\mathrm{ex}_0 m - 2| \times \\ &\times |\mathrm{ex}_0 m - 3|), \end{aligned}$$

$$H(m, y, z, u) = (z+u)\,\overline{\mathrm{sg}}\,|\mathrm{ex}_0 m - 1| + y\,\overline{\mathrm{sg}}\,|\mathrm{ex}_0 m - 2|.$$

Но условия (12) совпадают со схемой рекурсии 2-й ступени,

рассмотренной в предыдущем разделе. Из (11) видно, что функции $f_1(x)$, $f_2(x)$ удовлетворяют требованиям

$$f_1(x) \leqslant x, \quad f_2(x) \leqslant x,$$

наложенным на эти функции в п. 5.1. Применяя теорему о рекурсии 2-й ступени из п. 5.1, мы приходим к заключению, что функция $D(n, s)$, удовлетворяющая требованиям (8), (9), (10), не только существует, но и *является рекурсивной функцией*.

Теперь мы индукцией по параметру m докажем, что

$$D(m, x) = F(m, x) \qquad (x = 0, 1, 2, \ldots) \qquad (13)$$

для тех m, которые являются номером некоторого терма, и что для остальных значений m одноместная функция $D(m, x)$ все же является примитивно рекурсивной, хотя равенство (13) ввиду неопределенности правой части и не будет верным.

В самом деле, для начальных значений $m = 0, 1, 2, 3$ из (8), (9), (10) получаем непосредственно

$$D(0, x) = D(2, x) = 0, \quad D(1, x) = \boldsymbol{s}(x), \quad D(3, x) = \boldsymbol{q}(x).$$

Далее, пусть доказываемое утверждение справедливо для всех значений m, не превосходящих некоторого n. Рассмотрим выражение $D(n+1, x)$ как функцию от x. По условию, выражения $D(f_1(n), x)$, $D(f_2(n), x)$ являются примитивно рекурсивными функциями от x. Первые два равенства из системы (12) показывают, что функция $D(n+1, x)$ возникает из примитивно рекурсивных функций посредством обычной примитивной рекурсии и потому эта функция примитивно рекурсивная.

Наконец, пусть $n+1$ есть номер некоторого терма. Тогда $f_1(n)$ и $f_2(n)$ — также номера термов и, следовательно,

$$D(f_i(n), x) = F(f_i(n), x) \qquad (i = 1, 2;\ x = 0, 1, \ldots). \qquad (14)$$

Первые два равенства системы (12) для рассматриваемого значения $n+1$ справедливы как для функции $D(n+1, x)$, так и для функции $F(n+1, x)$. В силу (14) правые части указанных равенств одинаковы для функции $D(n+1, x)$ и для функции $F(n+1, x)$. Поэтому $D(n+1, x) = F(n+1, x)$, что и требовалось.

Итак, мы построили общерекурсивную функцию $D(n, x)$, обладающую следующими свойствами:

а) для каждого фиксированного n одноместная функция $D(n, x)$ является примитивно рекурсивной;

б) для каждой одноместной примитивно рекурсивной функции $f(x)$ существует такое число n (равное номеру терма, представляющего функцию f), что $D(n, x) = f(x)$.

Это означает, что $D(n, x)$ есть искомая общерекурсивная универсальная функция для класса всех одноместных примитивно рекурсивных функций.

Следствие 1. *Класс общерекурсивных функций шире класса примитивно рекурсивных функций: существуют общерекурсивные функции, не являющиеся примитивно рекурсивными.*

В качестве примера можно взять двуместную функцию $D(n, x)$. Она общерекурсивная и согласно теореме 1 примитивно рекурсивной быть не может. Из доказательства теоремы 1 видно, что общерекурсивная одноместная функция $D(x, x)$ также не является примитивно рекурсивной.

Следствие 2. *Для каждого* $n = 1, 2, \ldots$ *класс всех n-местных примитивно рекурсивных функций имеет общерекурсивную универсальную функцию.*

Для $n = 1$ искомой универсальной функцией является $D(n, x)$. Поэтому пусть $n \geqslant 2$. Рассмотрим функцию

$$D^{n+1}(x_0, x_1, \ldots, x_n) = D(x_0, c(x_1, \ldots, x_n)),$$

где $c(x_1, \ldots, x_n)$ — канторовский номер (п. 3.3) n-ки чисел $\langle x_1, \ldots, x_n \rangle$. Для любого фиксированного x_0 функция D^{n+1} является примитивно рекурсивной функцией от $x_1, \ldots, x_n$. С другой стороны, пусть $g(x_1, \ldots, x_n)$ — какая-нибудь n-местная примитивно рекурсивная функция. Тогда примитивно рекурсивной будет и одноместная функция

$$f(x) = g(c_{n1}(x), \ldots, c_{nn}(x)),$$

где $c_{ni}(x)$ есть i-й член n-ки, имеющий канторовский номер x (п. 3.3). Из универсальности функции D вытекает, что для подходящего x_0 для всех x будем иметь

$$f(x) = D(x_0, x).$$

Подставляя сюда вместо x число $c(x_1, \ldots, x_n)$ и замечая,

что

$$c_{ni}(c(x_1, \ldots, x_n)) = x_i,$$

получим

$$g(x_1, \ldots, x_n) = D(x_0, c(x_1, \ldots, x_n)).$$

Следовательно, $D^{n+1}(x_0, x_1, \ldots, x_n)$ — искомая общерекурсивная функция, универсальная для семейства всех примитивно рекурсивных n-местных функций.

Построив общерекурсивные функции, не являющиеся примитивно рекурсивными, естественно спросить себя, а существуют ли рекурсивные множества, не являющиеся примитивно рекурсивными (п. 4.1)? Ответ утвердительный. Чтобы найти такое множество, достаточно найти общерекурсивную функцию, принимающую лишь значения 0,1 и не совпадающую ни с одной примитивно рекурсивной функцией. Но такой функцией заведомо является функция

$$D(x) = \overline{\mathrm{sg}}\, D(x, x).$$

В самом деле, функция $D(x)$ общерекурсивная и принимает лишь значения 0,1. Если бы $D(x)$ была примитивно рекурсивной, то нашлось бы такое число n, что для всех x мы имели бы

$$\overline{\mathrm{sg}}\, D(x, x) = D(n, x),$$

откуда при $x = n$ получили бы противоречие

$$\overline{\mathrm{sg}}\, D(n, n) = D(n, n).$$

Множество, характеристической функцией которого служит функция $D(x)$, и есть искомое рекурсивное, но не примитивно рекурсивное множество.

Аналогичный вопрос о существовании нерекурсивных рекурсивно перечислимых множеств будет решен в п. 6.3.

5.3. Быстрорастущие функции. Мы решили задачу построения общерекурсивной функции, не являющейся примитивно рекурсивной, методом универсальных функций. Другим методом для решения той же задачи может служить метод построения функции, растущей быстрее любой функции заданного класса. Этот метод очень удобен при исследовании сравнительной силы различного рода рекурсий. Мы сейчас изложим его применительно к уже рассмотренному классу примитивно рекурсивных функций.

Итак, мы хотим найти по возможности простые, но очень быстрорастущие функции. Опыт показывает, что произведение растет быстрее суммы, степень быстрее произведения. Называя сложение, умножение и возведение в степень действиями 0-й, 1-й и 2-й ступени и вводя для них в целях единообразия обозначения

$$P_0(a, x) = a + x, \quad P_1(a, x) = ax, \quad P_2(a, x) = a^x,$$

приходим к знакомой всем идее о продолжении этой последовательности путем введения действий высших ступеней. При этом действие высшей ступени должно возникать из действия предыдущей ступени так же, как умножение возникает из сложения, возведение в степень из умножения. Функции P_0, P_1, P_2 связаны следующими соотношениями:

$$\begin{aligned} &P_1(a, x+1) = P_0(a, P_1(a, x)), \qquad P_1(a, 1) = a,\\ &P_2(a, x+1) = P_1(a, P_2(a, x)), \qquad P_2(a, 1) = a. \end{aligned} \tag{1}$$

Продолжим эту цепочку, полагая по определению для $n = 2, 3, \ldots$

$$P_{n+1}(a, 1) = a, \tag{2}$$

$$P_{n+1}(a, x+1) = P_n(a, P_{n+1}(a, x)). \tag{3}$$

Чтобы функции $P_n(a, x)$ были определены всюду, положим

$$P_{n+1}(a, 0) = 1 \qquad (n = 1, 2, \ldots)$$

и соотношения (2), (3) будем считать определениями функций $P_n(a, x)$ для $n = 2, 3, \ldots$ Ясно, что равенства (1) вытекают из соотношений (2), (3) и соотношения $P_1(a, 0) = 0$. В силу (2), (3), например, имеем

$$P_3(a, 0) = 1, \quad P_3(a, 1) = a, \quad P_3(a, 2) = a^a, \quad P_3(a, 3) = a^{a^a}.$$

Вводим новые функции

$$B(n, x) = P_n(2, x), \qquad A(x) = B(x, x).$$

Функции $B(n, x)$ часто называют функциями Аккермана, а функцию $A(x)$ — диагональной функцией Аккермана. Покажем, что функция $A(x)$ общерекурсивная.

Для функции $B(n, x)$ из соотношений (2), (3) вытекают следующие тождества:

$$B(n+1, x+1) = B(n, B(n+1, x)), \tag{4}$$

$$B(n+1, 0) = \operatorname{sg} n, \tag{5}$$

$$B(0, x) = 2 + x. \tag{6}$$

Сравнивая эти тождества со схемой рекурсии 2-й ступени, видим, что функция $B(n, x)$ возникает из примитивно рекурсивных функций при помощи рекурсии 2-й ступени. Следовательно, функция $B(n, x)$, а вместе с нею и функция $A(x)$, общерекурсивны.

Наша цель теперь состоит в доказательстве следующего предложения:

Теорема 1 (Аккерман [4]). *Для каждой одноместной примитивно рекурсивной функции* $f(x)$ *существует такое число* a, *что*

$$f(x) < A(x) \qquad (x = a,\ a+1,\ \ldots),$$

и, следовательно, функция $A(x)$ *не примитивно рекурсивная.*

Прежде чем переходить к доказательству этой теоремы, выведем несколько свойств функции $B(n, x)$.

$$\text{а) } B(n, x) \geqslant 2^x \qquad (n \geqslant 2;\ x = 1, 2, \ldots).$$

Так как $B(2, x) = 2^x$, то для $n = 2$ это соотношение верно. Далее применяем индукцию по n. Пусть для некоторого $n \geqslant 2$ соотношение а) истинно при всех значениях $x \geqslant 1$. Согласно (2)

$$B(n+1,\ 1) = P_{n+1}(2,1) = 2 \geqslant 2^1.$$

Пусть теперь для какого-то $x \geqslant 1$ имеем $B(n+1,\ x) \geqslant 2^x$. Тогда

$$B(n+1,\ x+1) = B(n,\ B(n+1,\ x)) \geqslant 2^{B(n+1,\ x)} \geqslant 2^{2^x} \geqslant 2^{x+1}$$

и, следовательно, соотношение а) доказано.

$$\text{б) } B(n,\ x+1) > B(n,\ x) \qquad (n,\ x = 1, 2, \ldots).$$

Так как $B(1, x) = 2x$, то для $n = 1$ это неравенство истинно. Пусть б) истинно для некоторого $n \geqslant 1$. В силу (4) и а)

$$B(n+1,\ x+1) = B(n,\ B(n+1,\ x)) \geqslant 2^{B(n+1,\ x)} > B(n+1,\ x),$$

что и требовалось.

$$\text{в) } B(n+1,\ x) \geqslant B(n,\ x+1) \qquad (n \geqslant 1,\ x \geqslant 2).$$

Действительно, в силу (4), а), б) имеем

$$B(n+1,\ x+1) = B(n,\ B(n+1,\ x)) \geqslant B(n,\ 2^x) \geqslant B(n,\ x+2).$$

Произвольную одноместную функцию $f(x)$ условимся называть B-мажорируемой, если существует такое n, что

$$f(x) < B(n,\ x) \qquad (x = 2, 3, \ldots).$$

Лемма. *Все одноместные примитивно рекурсивные функции B-мажорируемы.*

Доказательство этой леммы удобно разбить на ряд отдельных утверждений.

1) *Функции* $s(x) = x + 1$ *и* $q(x) = x - [\sqrt{x}]^2$ *B-мажорируемы*, так как для $x \geqslant 2$

$$q(x) < s(x) < 2^x = B(2,\ x).$$

2) *Сумма B-мажорируемых функций B-мажорируема.* Пусть произвольные функции $f(x)$, $g(x)$ для $x \geqslant 2$ удовлетворяют соотношениям

$$f(x) < B(n_1,\ x), \quad g(x) < B(n_2,\ x).$$

Тогда, полагая $n = n_1 + n_2$, имеем

$$f(x) < B(n,\ x), \qquad g(x) < B(n,\ x). \tag{7}$$

Отсюда согласно а), б), в) получаем

$$f(x)+g(x)<2B(n,\ x)\leqslant 2^{B(n+1,\ x)}\leqslant B(n,\ B(n+1,\ x))=$$
$$=B(n+1,\ x+1)\leqslant B(n+2,\ x),$$

что и требовалось.

3) *Суперпозиция В-мажорируемых функций В-мажорируема.* Действительно, из (7) последовательно получаем

$$f(g(x))<B(n,\ g(x))<B(n,\ B(n+1,\ x))=$$
$$=B(n+1,\ x+1)\leqslant B(n+2,\ x).$$

4) *Итерация В-мажорируемой функции В-мажорируема.* Пусть для $x\geqslant 2$ имеем $f(x)<B(n,\ x)$. Положим

$$m=n+f(0)+f(1)+f(f(0)).$$

Итерация $g(x)$ функции $f(x)$ определяется равенствами

$$g(0)=0,\qquad g(x+1)=f(g(x)).$$

Покажем, что

$$g(x)<B(m,\ x)\qquad (x\geqslant 2).\tag{8}$$

Для $x=2$ имеем

$$g(2)=f(f(0))<2^{f(f(0))+1}\leqslant B(n,\ f(f(0))+1)\leqslant B(m,\ 2).$$

Далее, по индукции предположим, что для некоторого $x\geqslant 2$ неравенство (8) справедливо. Рассмотрим $g(x+1)$. Возможны два случая: α) $g(x)\geqslant 2$ и β) $g(x)=0,\ 1$. В случае α) имеем

$$g(x+1)=f(g(x))<B(x,\ g(x))\leqslant B(m-1,\ B(m,x))=B(m,\ x+1).$$

В случае β) снова имеем

$$g(x+1)=f(g(x))<f(0)+f(1)<2^{m-n}\leqslant$$
$$\leqslant B(n,\ m-n+x+1)\leqslant B(m,\ x+1).$$

Таким образом, утверждение 4) доказано.

Теперь для вывода леммы остается лишь сослаться на теорему Робинсона из п. 3.5. Согласно этой теореме все примитивно рекурсивные функции получаются из функций $s(x)$ и $q(x)$ сложениями, суперпозициями и итерациями. Но согласно 1) функции s, q B-мажорируемы. Согласно 2), 3), 4) операции сложения, суперпозиции и итерации, будучи применены к B-мажорируемым функциям, дают снова B-мажорируемые функции. Поэтому все примитивно рекурсивные функции B-мажорируемые.

Докажем, наконец, теорему 1. Пусть $f(x)$ — некоторая примитивно рекурсивная функция. Согласно лемме для подходящего n при всех $x\geqslant 2$ будем иметь $f(x)<B(n,\ x)$. Но тогда

$$A(n+x)=B(n+x,\ n+x)\geqslant B(n,\ n+x)>f(n+x)\qquad (x\geqslant 2),$$

что и требовалось.

5.4. Обращение функций. Алгебра Робинсон. Согласно основному определению (п. 2.4) всюду определенная одноместная функция $f(x)$ называется общерекурсивной, если ее можно получить из базисных функций $s(x) = x + 1$, o, I_i^n с помощью конечного числа операций подстановок, примитивных рекурсий и μ-операций. Таким образом, для решения вопроса о рекурсивности одноместной функции $f(x)$ мы должны строить $f(x)$ из многоместных функций. Естественно возникает вопрос, а нельзя ли указать такие операции и такие одноместные исходные функции, чтобы все одноместные общерекурсивные функции и только их можно было получить из исходных, оставаясь все время в классе одноместных функций? Аналогичный вопрос уже ставился и был решен в п. 3.5 применительно к классу одноместных примитивно рекурсивных функций. Теперь мы рассмотрим указанный вопрос для общерекурсивных функций.

Пусть $f(x)$ — произвольная одноместная частичная функция. Согласно п. 2.3 функция $g(x)$, определенная формулой

$$g(x) = \mu_y (f(y) = x),$$

называется обратной для $f(x)$ и обозначается символически через f^{-1}. Оператор, ставящий в соответствие функции f функцию f^{-1}, называется *оператором обращения*. Легко видеть, что обратная функция $f^{-1}(x)$ будет тогда и только тогда всюду определенной, когда заданная функция $f(x)$ всюду определена и совокупность ее значений совпадает со всем натуральным рядом.

Основной нашей целью сейчас является доказательство следующего результата.

Теорема 1 (Юлия Робинсон [87]). *Каждая общерекурсивная одноместная функция может быть получена из функций $s(x) = x + 1$ и $q(x) = x - [\sqrt{x}]^2$ конечным числом операций сложения, суперпозиции и обращения одноместных функций. При этом операция обращения выполняется только тогда, когда результатом ее является всюду определенная функция.*

Доказательство этой теоремы в целом аналогично доказательству соответствующей теоремы Робинсона о примитивно рекурсивных функциях из п. 3.5. Так же как и в п. 3.5, одноместную функцию $f(x)$ назовем допустимой, если она может быть получена из функций s, q так, как указано в теореме 1. Многоместную функцию $F(x_1, \ldots, x_n)$ мы будем называть допустимой, если для любых одноместных допустимых функций $f_1(x), \ldots, f_n(x)$ будет допустимой одноместная функция $F(f_1(x), \ldots, f_n(x))$.

Теорема 1 будет доказана, если нам удастся доказать, что все общерекурсивные функции (как одноместные, так и многоместные) допустимы. Доказательство будет разбито на ряд пунктов, из которых затем будет выведено окончательное утверждение.

А) *Суперпозиция (многоместных) допустимых функций есть функция допустимая.* Доказательство очевидно.

Б) *Функция $q^{-1}(x)$ всюду определенная и*

$$q^{-1}(2x) = x^2 + 2x, \quad q^{-1}(2x+1) = x^2 + 4x + 2. \tag{1}$$

В самом деле, для любого x

$$x^2 \leqslant x^2+2x < (x+1)^2 < x^2+4x+2 < (x+2)^2$$

и, значит,

$$q(x^2+2x)=2x, \qquad q(x^2+4x+2)=2x+1.$$

Таким образом, числа

$$y=y_0=x^2+2x, \qquad z=z_0=x^2+4x+2$$

являются решениями уравнений

$$q(y)=2x, \qquad q(z)=2x+1. \tag{2}$$

Надо показать, что y_0, z_0 — наименьшие решения. Пусть y, z — какие-нибудь решения уравнений (2). Это значит, что

$$y=m^2+2x<(m+1)^2, \qquad z=n^2+2x+1<(n+1)^2.$$

Отсюда следует, что $x \leqslant m$, $n \geqslant x+1$ и потому

$$y=m^2+2x \geqslant x^2+2x=y_0,$$
$$z=n^2+2x+1 \geqslant (x+1)^2+2x+1=z_0.$$

Таким образом, формулы (1) истинны.

В) *Функции* x, o, $x+y$, $I_i^n(x_1, \ldots, x_n)=x_i$ *допустимы.*

Это непосредственно вытекает из очевидных формул

$$q(q^{-1}(x))=x, \qquad q(q^{-1}(x+x)+1)=o$$

и определения допустимости функций.

Г) *Функции вида* $ax+by+c$, *где* a, b, c — *произвольные фиксированные натуральные числа, допустимы.*

В силу А), В) функции $I_1^2(x, y)=x$, $I_2^2(x, y)=y$, $s(0)=1$ допустимы. Функция $ax+by+c$ возникает из упомянутых функций повторными сложениями и потому сама допустима.

Д) *Функции* x^2, $\operatorname{sg} x$, $\overline{\operatorname{sg}}\, x$, $[x/2]$, xy, $x \dot{-} y$, $[\sqrt{x}]$ *допустимы.*

В п. 3.5 была введена функция

$$x \div y=q((x+y)^2+5x+3y+4)=\begin{cases} x-y, & x \geqslant y, \\ 3x+y+3, & x<y. \end{cases} \tag{3}$$

В силу Б)

$$(x+y)^2+5x+3y+4=q^{-1}(2x+2y)+3x+y+4$$

и потому на основании Г) функция $x \div y$ допустима.

Теперь из очевидных формул

$$x^2=q^{-1}(2x) \div 2x,$$
$$\operatorname{sg} x=q(x^2+1),$$
$$\overline{\operatorname{sg}}\, x=1 \div \operatorname{sg} x$$

заключаем, что функции x^2, $\operatorname{sg} x$, $\overline{\operatorname{sg}}\, x$ допустимы.

Более сложно выводится допустимость $[x/2]$. Пусть

$$f(x) = \overline{\text{sg}}\,(q\,(q^{-1}(x) + 2)). \tag{4}$$

Для $x = 2t$ согласно Б) имеем

$$f(x) = \overline{\text{sg}}\,(q\,(t^2 + 2t + 2)) = 0.$$

Если же $x = 2t + 1$, то

$$f(x) = \overline{\text{sg}}\,(q\,(t^2 + 4t + 4)) = 1.$$

Иначе говоря, $f(x)$ есть характеристическая функция для свойства быть четным числом и формула (4) лишь показывает, что эта функция допустима. Рассмотрим уравнение

$$f(y) + 2q(y) = x. \tag{5}$$

Полагая $y_0 = (x + [x/2])^2 + [x/2]$ и принимая во внимание соотношение $f(z^2) = f(z)$, получим $q(y_0) = [x/2]$, $f(y_0) = f(x)$ и, значит,

$$2q(y_0) + f(y_0) = 2[x/2] + f(x) = x.$$

Это показывает, что уравнение (5) имеет решение для каждого x. Поэтому функция $(2q + f)^{-1}$ всюду определенная, и, следовательно, допустимая. Из (5) вытекает, что

$$q(y) = \frac{1}{2}(x \div f(y)) \tag{6}$$

для любого y, удовлетворяющего соотношению (5). В частности, (6) должно быть верным и для $y = (2q + f)^{-1}(x)$, то есть

$$q((2q + f)^{-1}(x)) = \frac{1}{2}(x \div f(x)) = [x/2].$$

Следовательно, функция $[x/2]$ допустима.

Из очевидного соотношения

$$xy = \left[\frac{((x + y)^2 \div x^2) \div y^2}{2}\right]$$

заключаем, что допустима и функция xy.

Из формулы (3) вытекает, что

$$((x \div y) + y) \div x = \begin{cases} 0, & x \geqslant y, \\ 2x + 2y + 3, & x < y. \end{cases}$$

Поэтому, вводя функцию

$$v(x, y) = \overline{\text{sg}}\,((x \div y) + y \div x) = \begin{cases} 1, & x \geqslant y, \\ 0, & x < y, \end{cases}$$

будем иметь

$$x \dot{-} y = v(x, y)(x \div y).$$

Следовательно, функция $x \dot{-} y$ допустима.

Наконец, рассмотрим выражение

$$w(x) = \left[\frac{1}{2} q(x \dot{-} 1)\right] + \operatorname{sg} x. \qquad (7)$$

Для $x = 0$ имеем $w(0) = 0$. Если же $x = t^2$, $t > 0$, то

$$w(t^2) = \left[\frac{1}{2} q((t-1)^2 + 2t - 2)\right] + 1 = \frac{1}{2}(2t-2) + 1 = t.$$

Следовательно, для любого t имеем $w(t^2) = t$ и потому

$$w(x \dot{-} q(x)) = w([\sqrt{x}]^2) = [\sqrt{x}] \qquad (8)$$

для каждого x. Формулы (7), (8) показывают, что функция $[\sqrt{x}]$ допустима.

Е) *Канторовские функции* (п. 3.3) $c(x, y)$, $l(x)$, $r(x)$ *допустимы*. Эти функции получаются из функций 1, x, y с помощью операций $+$, $\dot{-}$, $[\sqrt{\ }]$, $[/2]$, которые согласно доказанному не выводят за пределы класса допустимых функций.

Ж) *Если функция* $h(x, y)$ *допустима и уравнение* $h(x, y) = 0$ *разрешимо для каждого* x, *то функция*

$$f(x) = \mu_y(h(x, y) = 0) \qquad (9)$$

также допустима.

Докажем сначала, что

$$r(\mu_z((l(z)+1)\,\overline{\operatorname{sg}}\, h(l(z), r(z)) = x+1)) = f(x). \qquad (10)$$

Действительно, уравнение

$$(l(z)+1)\,\overline{\operatorname{sg}}\,(h(l(z), r(z))) = x+1$$

равносильно системе

$$l(z) = x, \qquad h(x, r(z)) = 0. \qquad (11)$$

Так как по условию $h(x, f(x)) = 0$ и, кроме того,

$$l(c(u, v)) = u, \quad r(c(u, v)) = v,$$

то $z_0 = c(x, f(x))$ есть решение системы (11). Надо лишь проверить, является ли z_0 наименьшим решением этой системы.

Пусть z — произвольное решение системы (11). Из второго уравнения этой системы следует

$$r(z) \geqslant f(x) = r(z_0), \qquad (12)$$

а из первого получаем, что

$$l(z) = l(z_0). \qquad (13)$$

Согласно п. 3.3 для пар натуральных чисел, имеющих фиксированный первый элемент, канторовский номер пары растет вместе с ростом второго элемента. Поэтому из (12) и (13) вытекает, что $z_0 \leqslant z$. Следовательно,

$$\mu_z((l(z)+1)\,\overline{\operatorname{sg}}\, h(l(z), r(z)) = x+1) = z_0$$

и потому соотношение (10) истинно. Но соотношение (10) показывает, что $f(x)$ есть суперпозиция допустимой функции $\boldsymbol{r}$ и обращения допустимой функции

$$(\boldsymbol{l}(z)+1)\,\overline{\mathrm{sg}}\,h(\boldsymbol{l}(z),\ \boldsymbol{r}(z)) \dot{-} 1.$$

Поэтому функция $f(x)$ допустимая.

З) *Функция* $[x/y]$ *и функция Геделя* $\boldsymbol{\Gamma}(x, y)$ (п. 3.3) *суть функции допустимые.*

Это непосредственно вытекает из предыдущих результатов в силу формул из пп. 3.2 и 3.3, посредством которых определяются $[x/y]$ и $\boldsymbol{\Gamma}(x, y)$.

Доказательство теоремы Робинсон на этом можно считать законченным. Действительно, в п. 3.4 было доказано, что каждую общерекурсивную функцию можно получить из простейших функций $\boldsymbol{o}$, $\boldsymbol{s}$, I_m^n и функций $+$, $\dot{-}$, $\boldsymbol{\Gamma}$, c, $\boldsymbol{l}$, $\boldsymbol{r}$ конечным числом операций подстановки и специальной минимизации вида (9). Но в пунктах Д), Е), З) показано, что все упомянутые функции допустимые, а в пунктах А), Ж) показано, что подстановки и специальные минимизации, примененные к допустимым функциям, дают функции допустимые. Таким образом, все общерекурсивные функции допустимые.

Аналогично теореме Робинсона (п. 3.5), теорема Ю. Робинсон также допускает простое алгебраическое истолкование. Совокупность $\mathfrak{F}_l^1$ всех всюду определенных и совокупность $\mathfrak{F}_{\mathrm{g.\,r}}^1$ всех общерекурсивных одноместных функций, рассматриваемых вместе с определенными на них операциями сложения $+$, композиции $*$ и обращения $^{-1}$, образуют две алгебры: алгебру Робинсон

$$\mathfrak{R}_l = \langle \mathfrak{F}_l^1;\ +,\ *,^{-1}\rangle$$

всех одноместных функций и алгебру Робинсон

$$\mathfrak{R}_{\mathrm{g.\,r}} = \langle \mathfrak{F}_{\mathrm{g.\,r}}^1;\ +,\ *,\ ^{-1}\rangle$$

всех одноместных общерекурсивных функций. Операция $^{-1}$, примененная к всюду определенной функции, дает, вообще говоря, частичную функцию, не входящую в $\mathfrak{F}_l^1$. Поэтому алгебры $\mathfrak{R}_l$ и $\mathfrak{R}_{\mathrm{g.\,r}}$ частичные. Теорема Робинсон утверждает, что элементы $\boldsymbol{s}$, $\boldsymbol{q}$ в алгебре $\mathfrak{R}_l$ порождают подалгебру $\mathfrak{R}_{\mathrm{g.\,r}}$.

Дополнения, примеры и задачи

1. Пусть $D(n, x)$ — универсальная функция, построенная в п. 5.2. Если $f(x) = D(n, x)$, то n называется D-номером $f(x)$. Показать, что каждая примитивно рекурсивная функция имеет бесконечно много D-номеров.

2. Показать, что существуют примитивно рекурсивные функции $f(m, n)$, $g(m, n)$, $h(n)$, удовлетворяющие тождествам

$$D(m, x) + D(n, x) = D(f(m, n), x),$$
$$D(m, D(n, x)) = D(g(m, n), x),$$
$$D^J(n, x) = D(h(n), x),$$

то есть позволяющие по D-номерам m, n каких-то примитивно рекурсивных одноместных функций находить номер их суммы, композиции, итерирования.

3. Функции, которые можно получить из простейших функций o, s, I_m^n операциями подстановки, примитивной рекурсии и рекурсии 2-й ступени, назовем функциями 2-й ступени. Пользуясь методом быстрорастущих функций, показать, что существуют общерекурсивные функции, не являющиеся функциями 2-й ступени (см. Р. Петер [67]).

4. Пусть $\mathfrak{F}^1_{\text{p. r}}$ — совокупность всех одноместных частично рекурсивных функций. Операция обращения $^{-1}$ будет всюду определенной на $\mathfrak{F}^1_{\text{p. r}}$. Показать, что элементы s, q являются порождающими алгебры $\langle \mathfrak{F}_{\text{p. r}}; +, *, ^{-1} \rangle$ (см. п. 5.4).

5. Показать, что алгебра $\langle \mathfrak{F}_{\text{g}}{}^1{}_{\text{. r}}; *, ^{-1} \rangle$ порождается двумя подходящими элементами и не порождается никаким одним элементом. Функции $c\,(lr\,(x), rl\,(x))$, $c\,(x, 1)$, $c\,(l\,(x) + lr\,(x), x)$ заведомо порождают указанную алгебру (Ю. Робинсон [87]).

6. Пусть $\mathfrak{F}^2$ — совокупность всех двуместных частичных функций, $\mathfrak{F}^2_{\mathfrak{S}}$ — совокупность всех двуместных частичных функций, частично рекурсивных относительно какой-то фиксированной системы $\mathfrak{S}$ частичных функций. Показать, что $\mathfrak{F}^2_{\mathfrak{S}}$ порождается с помощью операций $+$, $*$, M системой $\mathfrak{S}$ и функциями I_1^2, I_2^2, $s\,(I_2^2)$, $q\,(I_2^2)$ (M — операция минимизации, определенная в п. 2.3, операция $*$ для двуместных функций определена в задаче 7 к § 3).

7. Верно ли следующее утверждение: в алгебре $\langle \mathfrak{F}^1; +, *, ^{-1} \rangle$, где $\mathfrak{F}^1$ — совокупность всех одноместных частичных функций, произвольная система $\mathfrak{S} \subseteq \mathfrak{F}^1$, содержащая функции s, q, порождает подалгебру, состоящую из всех одноместных частичных функций, частично рекурсивных относительно $\mathfrak{S}$?

8. Показать, что совокупность всех одноместных функций 2-й ступени (см. выше задачу 3) обладает общерекурсивной универсальной функцией (см. Р. Петер [67]).

§ 6. ЧАСТИЧНО РЕКУРСИВНЫЕ ФУНКЦИИ

В этом параграфе сначала показывается, что каждая частично рекурсивная функция допускает параметрическое представление при помощи примитивно рекурсивных функций, а затем с помощью построенной в п. 5.2 рекурсивной универсальной функции $D\,(n, x)$ строится частично рекурсивная функция, универсальная для всех одноместных частично рекурсивных функций. Это центральный результат теории частично рекурсивных функций. С помощью частично рекурсивной универсальной функции строится пример нерекурсивного рекурсивно перечислимого

множества. В конце параграфа рассматриваются некоторые специальные формы, к которым могут быть приведены частично рекурсивные функции.

6.1. Параметризация частично рекурсивных функций. Согласно п. 4.4 графиком n-местной частичной функции $f(x_1, \ldots, x_n)$ называется совокупность последовательностей $\langle x_1, \ldots, x_n, y\rangle$ натуральных чисел, удовлетворяющих соотношению

$$f(x_1, \ldots, x_n) = y.$$

В частности, график нигде не определенной функции есть пустое множество.

Теорема 1 (о графике частично рекурсивной функции). *Для того чтобы частичная функция f была частично рекурсивной, необходимо и достаточно, чтобы график f был рекурсивно перечислим.*

Для нигде не определенной функции f утверждение этой теоремы очевидно. Для функций с непустым графиком утверждение теоремы 1, как легко видеть, означает, что функция $f(x_1, \ldots, x_n)$ тогда и только тогда частично рекурсивная, когда она представима в параметрической форме

$$x_i = \alpha_i(t) \qquad (i = 1, \ldots, n), \qquad y = \beta(t),$$

где $\alpha_i(t)$, $\beta(t)$ — подходящие примитивно рекурсивные функции. По этой причине теорему 1 иногда называют теоремой о параметрическом представлении частично рекурсивных функций.

Достаточность условий теоремы 1 устанавливается так же, как в п. 4.4. Доказательство необходимости удобно разбить на ряд самостоятельных утверждений. А именно, мы докажем последовательно, что операции подстановки, примитивной рекурсии и минимизации, примененные к функциям, имеющим рекурсивно перечислимый график, дают функции с рекурсивно перечислимым графиком. Так как все частично рекурсивные функции получаются посредством указанных операций из простейших функций $\boldsymbol{o}$, $\boldsymbol{s}$, I^k_m, имеющих согласно п. 4.4 примитивно рекурсивный график, то тем самым будет доказана и теорема 1. Более того, упомянутые утверждения относительно операций примитивной рекурсии и минимизации нам нет необходи-

мости доказывать в общем виде. В силу результатов п. 3.4 достаточно будет рассматривать примитивные рекурсии и минимизации тех частных видов, которые указаны в п. 3.4.

А) *Суперпозиция функций, имеющих рекурсивно перечислимые графики, есть функция с рекурсивно перечислимым графиком.*

Оставляя в стороне тривиальный случай, когда какая-нибудь из заданных функций нигде не определена, предположим, что заданные функции g, g_1, ..., g_m имеют графики

$$G = \{\langle \alpha_1(t), \ldots, \alpha_{m+1}(t)\rangle\},$$
$$G_i = \{\langle \beta_{i1}(t), \ldots, \beta_{i\,n+1}(t)\rangle\} \qquad (i = 1, \ldots, m),$$

где α_i, β_{ij} — подходящие примитивно рекурсивные функции. Нам надо изучить график $\boldsymbol{F}$ функции

$$f(x_1, \ldots, x_n) = g(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n)).$$

Согласно определению операции суперпозиции утверждение $\langle x_1, \ldots, x_n, y\rangle \in \boldsymbol{F}$ можно представить в виде

$$(\exists y_1 \ldots y_m)(\langle x_1, \ldots, x_n, y_1\rangle \in \boldsymbol{G}_1 \& \ldots$$
$$\ldots \& \langle x_1, \ldots, x_n, y_m\rangle \in \boldsymbol{G}_m \& \langle y_1, \ldots, y_m, y\rangle \in \boldsymbol{G}), \qquad (1)$$

где $(\exists\, y_1 \ldots y_m)$ обозначает утверждение «*существуют такие* y_1, ..., y_m, *что* ...», а символ & обозначает слово *и* (см. список основных обозначений). В свою очередь, утверждение (1) с помощью графиков функций g, g_1, ..., g_m можно переписать в виде

$$(\exists t_0 t_1 \ldots t_m)(\beta_{11}(t_1) = x_1 \& \ldots \& \beta_{1n}(t_1) = x_n \& \ldots$$
$$\ldots \& \beta_{m1}(t_m) = x_1 \& \ldots \& \beta_{mn}(t_m) = x_n \&$$
$$\& \alpha_1(t_0) = \beta_{1\,n+1}(t_1) \& \ldots \& \alpha_m(t_0) = \beta_{m\,n+1}(t_m) \& \alpha_{m+1}(t_0) = y).$$

Поэтому, вводя функцию

$$h(x_1, \ldots, x_n, y, t_0, \ldots, t_m) =$$
$$= \sum_{i,\,j} |\beta_{ij}(t_i) - x_j| + \sum_i |\alpha_i(t_0) - \beta_{i\,n+1}(t_i)| + |\alpha_{m+1}(t_0) - y|,$$

получим равносильное (1) утверждение

$$(\exists t_0 \ldots t_m)(h(x_1, \ldots, x_n, y, t_0, \ldots, t_m) = 0).$$

Это означает, что график функции f состоит из всех тех последовательностей $\langle x_1, \ldots, x_n, y\rangle$, для которых уравнение

$$h(x_1, \ldots, x_n, y, t_0, \ldots, t_m) = 0$$

разрешимо относительно $t_0, \ldots, t_m$. Так как функция h примитивно рекурсивная, то на основании теоремы 1 из п. 4.4 заключаем, что график функции f рекурсивно перечислим.

Б) *Если функция $f(x, y)$ связана соотношениями*

$$f(x, 0) = x,$$
$$f(x, y+1) = h(f(x, y)) \tag{2}$$

с функцией h, имеющей рекурсивно перечислимый график, то $f(x, y)$ также имеет рекурсивно перечислимый график.

Оставляя снова в стороне тривиальный случай, когда график функции h пустой, мы предположим, что график h состоит из пар

$$\langle \alpha(t), \beta(t)\rangle \quad (t = 0, 1, \ldots),$$

где α, β — подходящие примитивно рекурсивные функции. График $\boldsymbol{F}$ функции f состоит из троек $\langle x, y, z\rangle$, для которых $f(x, y) = z$. Разобьем его на две части: совокупность $\boldsymbol{F}_1$ троек $\langle x, y, z\rangle$, для которых $y = 0$, и совокупность $\boldsymbol{F}_2$ троек $\langle x, y, z\rangle$, для которых $y > 0$. Первое из соотношений (2) показывает, что $\boldsymbol{F}_1$ состоит из всех троек вида $\langle x, 0, x\rangle$ и потому $\boldsymbol{F}_1$ — рекурсивно перечислимая совокупность. Остается показать, что совокупность $\boldsymbol{F}_2$ также рекурсивно перечислима.

Соотношение (2) показывает, что условие $\langle x, y, z\rangle \in \boldsymbol{F}$ равносильно существованию чисел $a_1, \ldots, a_{y \dot{-} 1}$ таких, что

$$a_1 = h(x), \quad a_2 = h(a_1), \ \ldots, \ z = h(a_{y \dot{-} 1}).$$

Равенство $a_{i+1} = h(a_i)$ эквивалентно существованию числа t_{i+1}, для которого

$$\alpha(t_{i+1}) = a_i, \quad \beta(t_{i+1}) = a_{i+1}.$$

Поэтому условие $\langle x, y, z\rangle \in \boldsymbol{F}_2$ равносильно существованию чисел $t_1, t_2, \ldots, t_y$, для которых

$$\alpha(t_1) = x, \ \alpha(t_2) = \beta(t_1), \ \ldots, \ \alpha(t_y) = \beta(t_{y \dot{-} 1}), \ \beta(t_y) = z,$$

то есть для которых

$$|\alpha(t_1)-x|+\sum_{i=1}^{y\dot{-}1}|\alpha(t_{i+1})-\beta(t_i)|+|\beta(t_y)-z|=0. \qquad (3)$$

В п. 3.3 была построена примитивно рекурсивная функция Геделя $\mathbf{\Gamma}(u, x)$ и было показано, что для любых $t_1, \ldots, t_y$ система уравнений

$$\mathbf{\Gamma}(u, i)=t_i \qquad (i=1, \ldots, y)$$

имеет для u решения. Подставляя в (3) вместо t_i выражение $\mathbf{\Gamma}(u, i)$, получим уравнение

$$|\alpha(\mathbf{\Gamma}(u, 1))-x|+\sum_{i=1}^{y\dot{-}1}|\alpha(\mathbf{\Gamma}(u, i+1))-\beta(\mathbf{\Gamma}(u, i))|+$$
$$+|\beta(\mathbf{\Gamma}(u, y))-z|=0, \qquad (4)$$

имеющее решение u тогда и только тогда, когда (3) имеет решение $(t_1, \ldots, t_y)$. Согласно п. 3.1 левая часть уравнения (4) есть примитивно рекурсивная функция от x, y, z, u. Обозначая ее через $g(x, y, z, u)$, приходим к заключению, что совокупность $\boldsymbol{F}_2$ состоит из всех тех троек $\langle x, y, z\rangle$, для которых уравнение

$$g(x, y, z, u)=0$$

имеет решение u. Так как g — функция примитивно рекурсивная, то множество $\boldsymbol{F}_2$ — рекурсивно перечислимое, что и требовалось.

В) *Если частичная функция* $g(x, y)$ *имеет рекурсивно перечислимый график, то функция* $f(x)$, *возникающая из* $g(x, y)$ *операцией минимизации*

$$f(x)=\mu_y(g(x, y)=0) \qquad (5)$$

также имеет рекурсивно перечислимый график.

Как и выше, мы можем предполагать, что график $\mathbf{G}$ функции g не пуст и, следовательно, имеет вид

$$\boldsymbol{G}=\{\langle\alpha(t), \beta(t), \gamma(t)\rangle\} \qquad (t=0, 1, \ldots),$$

где α, β, γ — примитивно рекурсивные функции. Соотношение (5) равносильно утверждению о существовании чисел $a_0, a_1, \ldots, a_y$, удовлетворяющих требованиям

$$g(x, i)=a_i \qquad (i=0, 1, \ldots, y),$$
$$a_j\neq 0, \quad a_y=0 \qquad (j=0, 1, \ldots, y-1),$$

то есть равносильно утверждению о существовании чисел $t_0, \ldots, t_y$, для которых

$$\alpha(t_j)=x,\ \beta(t_j)=j,\ \gamma(t_j)\neq 0 \quad (j=0,\ 1,\ \ldots,\ y-1),$$
$$\alpha(t_y)=x,\ \beta(t_y)=y,\ \gamma(t_y)=0. \tag{6}$$

Условия (6) равносильны соотношению

$$\sum_{i=0}^{y}(|\alpha(t_i)-x|+|\beta(t_i)-i|)+y\,\overline{\mathrm{sg}}\prod_{i=0}^{y\dot{-}1}\gamma(t_i)+\gamma(t_y)=0. \tag{7}$$

Полагая $t_i = \mathbf{\Gamma}(u,\ i)$, видим, что

$$\langle x,\ y\rangle\in F \Longleftrightarrow (\exists u)\,(h(x,\ y,\ u)=0), \tag{8}$$

где через $h(x,\ y,\ u)$ обозначено выражение, получающееся из левой части соотношения (7) подстановкой $\mathbf{\Gamma}(u,\ i)$ вместо t_i. Поскольку функция h примитивно рекурсивная, то из (8) следует, что график $\boldsymbol{F}$ функции f рекурсивно перечислим.

Теорема 1 доказана. Отметим несколько ее следствий.

Следствие 1. *Область определенности каждой частично рекурсивной функции есть множество, рекурсивно перечислимое.*

Действительно, если график частично рекурсивной функции f^s не пуст, то согласно теореме 1 его можно представить в виде

$$\{\langle u_1(t),\ \ldots,\ u_{s+1}(t)\rangle\},$$

где $u_1(t), \ldots, u_{s+1}(t)$ — примитивно рекурсивные функции. Но в таком случае областью определенности функции f будет совокупность s-ок вида $\langle u_1(t),\ \ldots,\ u_s(t)\rangle$ $(t = 0,\ 1,\ \ldots)$ и потому эта область будет рекурсивно перечислимой.

Следствие 2. *Совокупность значений, принимаемых частично рекурсивной функцией f, является рекурсивно перечислимой.*

В самом деле, если эта совокупность не пустая, то график функции можно представить в указанном виде $\langle u_1(t),\ \ldots,\ u_{s+1}(t)\rangle$ и совокупностью всех значений функции f будет совокупность всех значений функции $u_{s+1}(t)$. Так как функция u_{s+1} примитивно рекурсивна, то совокупность ее значений есть (п. 4.2) рекурсивно перечислимое множество.

Прежде чем сформулировать еще одно следствие, напомним разницу между *характеристической функцией* множества и *частичной характеристической функцией*: характеристической функцией множества $\boldsymbol{A}$ называется функция, равная 0 в точках $\boldsymbol{A}$ и равная 1 вне $\boldsymbol{A}$; частичной характеристической функцией $\boldsymbol{A}$ называется функция, равная 0 на $\boldsymbol{A}$ и не определенная вне $\boldsymbol{A}$.

Следствие 3. *Множество $\boldsymbol{A}$ n-ок чисел тогда и только тогда рекурсивно перечислимо, когда его частичная характеристическая функция частично рекурсивна.*

Пусть $\boldsymbol{A}$ — непустое рекурсивно перечислимое множество. Оно состоит из n-ок вида $\langle u_1(t), \ldots, u_n(t)\rangle$, где $u_i(t)$ примитивно рекурсивные функции. График частичной характеристической функции $\varphi(x_1, \ldots, x_n)$ множества $\boldsymbol{A}$ состоит из всех последовательностей вида $\langle u_1(t), \ldots, u_n(t), 0\rangle$, $t = 0, 1, \ldots$ Так как $0 = \boldsymbol{o}(t)$ есть функция примитивно рекурсивная, то график $\varphi(x_1, \ldots, x_n)$ рекурсивно перечислим, а сама функция φ частично рекурсивна.

Обратное утверждение непосредственно вытекает из следствия 1, так как каждое множество есть область определенности своей частичной характеристической функции.

Следствие 4. *Пусть $F(x_1, \ldots, x_m, y_1, \ldots, y_n)$ — частично рекурсивная функция от $m + n$ переменных. Тогда совокупность $\boldsymbol{A}$ тех m-ок $\langle x_1, \ldots, x_m\rangle$, для которых уравнение*

$$F(x_1, \ldots, x_m, y_1, \ldots, y_n) = 0$$

имеет хотя бы одно решение $\langle y_1, \ldots, y_n\rangle$, является рекурсивно перечислимым множеством.

Согласно теореме о параметризации график функции F, если он не пуст, можно представить в виде совокупности последовательностей

$$\langle \alpha_1(t), \ldots, \alpha_m(t), \alpha_{m+1}(t), \ldots, \alpha_{m+n}(t), \alpha(t)\rangle \quad (t = 0, 1, 2, \ldots),$$

где $\alpha_i(t)$, $\alpha(t)$ — подходящие примитивно рекурсивные функции. Совокупность $\boldsymbol{A}$ состоит из m-ок $\langle \alpha_1(t), \ldots$ $\ldots, \alpha_m(t)\rangle$, где t пробегает совокупность решений уравнения $\alpha(t) = 0$. Согласно п. 4.2 эта совокупность рекурсивно перечислима и потому либо пуста, либо является совокупностью значений подходящей примитивно рекурсивной

функции $\beta(u)$. Отсюда видно, что совокупность A либо пустая, либо состоит из всевозможных m-ок вида

$$\langle \alpha_1(\beta(u)), \ldots, \alpha_m(\beta(u))\rangle \qquad (u = 0, 1, 2, \ldots).$$

Так как функции $\alpha_i(\beta(u))$ примитивно рекурсивные, то совокупность A рекурсивно перечислима, что и требовалось доказать.

Следствие 5. *Совокупность последовательностей* $\langle x_1, \ldots, x_s\rangle$, *удовлетворяющих уравнению*

$$f(x_1, \ldots, x_s) = 0, \tag{9}$$

левая часть которого f есть частично рекурсивная функция от $x_1, \ldots, x_s$, является рекурсивно перечислимым множеством.

В самом деле, функция

$$\alpha(x) = \mu_t(x + t = 0)$$

частично рекурсивная и определенная только для $x = 0$. Следовательно, совокупность решений уравнения (9) совпадает с областью определенности частично рекурсивной функции $\alpha(f(x_1, \ldots, x_s))$. Согласно следствию 1 область определенности этой функции рекурсивно перечислима, что и требовалось.

6.2. Универсальные частично рекурсивные функции. Установленная в предыдущем разделе теорема о параметризации позволяет легко доказать следующую важную теорему Клини о нормальной форме.

Теорема 1. *Каждая частично рекурсивная функция $f(x_1, \ldots, x_s)$ представима в форме*

$$f(x_1, \ldots, x_s) = \boldsymbol{l}(\mu_t(F(x_1, \ldots, x_s, t) = 0)), \tag{1}$$

где $\boldsymbol{l}(x)$ — нумерационная функция из п. 3.3, а F — подходящая (зависящая от f) примитивно рекурсивная функция.

По условию график $\boldsymbol{M}$ функции f рекурсивно перечислим и, следовательно (п. 4.4), существует примитивно рекурсивная функция $g(x_1, \ldots, x_s, y, a)$, для которой

$$\langle x_1, \ldots, x_s, y\rangle \in \boldsymbol{M} \Leftrightarrow (\exists a)(g(x_1, \ldots, x_s, y, a) = 0).$$

Полагая $\boldsymbol{c}(y, a) = t$, видим, что

$$\langle x_1, \ldots, x_s, y\rangle \in \boldsymbol{M} \Leftrightarrow$$
$$\Leftrightarrow (\exists t)\,(g(x_1, \ldots, x_s, \boldsymbol{l}(t), \boldsymbol{r}(t)) = 0 \,\&\, y = \boldsymbol{l}(t)). \tag{2}$$

В частности, если для каких-нибудь $x_1, \ldots, x_s, t$

$$g(x_1, \ldots, x_s, \boldsymbol{l}(t), \boldsymbol{r}(t)) = 0,$$

то $\langle x_1, \ldots, x_s, \boldsymbol{l}(t) \rangle \in M$ и, значит, $\boldsymbol{l}(t) = f(x_1, \ldots, x_s)$, то есть

$$\boldsymbol{l}(\mu_t(g(x_1, \ldots, x_s, \boldsymbol{l}(t), \boldsymbol{r}(t)) = 0)) = f(x_1, \ldots, x_s). \quad (3)$$

Формула (3) доказана нами для тех значений $x_1, \ldots, x_s$, для которых левая часть равенства (3) имеет определенное значение. Но из (2) видно, что из неопределенности значения левой части формулы (3) вытекает неопределенность значения $f(x_1, \ldots, x_s)$. Поэтому равенство (3) справедливо для произвольных значений $x_1, \ldots, x_s$. Полагая

$$g(x_1, \ldots, x_s, \boldsymbol{l}(t), \boldsymbol{r}(t)) = F(x_1, \ldots, x_s, t),$$

получим формулу (1).

Замечание. Из доказательства формулы (1) видно, что в этой формуле вместо функции $\boldsymbol{l}(x)$ можно взять «левую» функцию $L(x)$ из любой тройки функций $L(x)$, $R(x)$, $C(x, y)$, связанных с какой-нибудь нумерацией пар натуральных чисел (см. п. 3.3), лишь бы функции $L(x)$, $R(x)$ были примитивно рекурсивными.

До сих пор мы различали всюду определенные частично рекурсивные функции (*рекурсивные* функции) и *общерекурсивные* функции, то есть функции, которые могут быть получены из простейших функций $\boldsymbol{o}$, $\boldsymbol{s}$, I_m^n операциями подстановки, примитивной рекурсии и операцией минимизации, производимой лишь над такими функциями, для которых результат минимизации является всюду определенной функцией. Нам было лишь известно, что каждая общерекурсивная функция рекурсивная. Обратное утверждение верно.

Следствие. *Каждая рекурсивная функция f общерекурсивна.*

В самом деле, согласно теореме 1 функция f может быть представлена в форме (1), которая показывает, что функция f получается из простейших функций операциями подстановки, примитивной рекурсии и операцией минимизации μ, примененной единственный раз к функции $F(x_1, \ldots, x_s, t)$. Так как функция f по условию всюду

определенная, то функция

$$\mu_t (F(x_1, \ldots, x_s, t) = 0)$$

также должна быть всюду определенной, что и требовалось.

Итак, понятия рекурсивной и общерекурсивной функции равносильны.

Рассмотрим теперь вопрос о построении универсальных функций. Пусть $\mathfrak{F}^s_{\text{p. r}}$ обозначает совокупность всех s-местных частично рекурсивных функций. Как уже говорилось в п. 5.2, частичная функция $U(x_0, x_1, \ldots, x_s)$ называется *универсальной* для совокупности $\mathfrak{F}^s_{\text{p. r}}$, если $\mathfrak{F}^s_{\text{p. r}}$ состоит из функций

$$U(0, x_1, \ldots, x_s), \quad U(1, x_1, \ldots, x_s), \ldots$$

В п. 5.2 была построена общерекурсивная функция $D^{s+1}(n, x_1, \ldots, x_s)$, универсальная для совокупности $\mathfrak{F}^s_{\text{pr. r}}$ всех примитивно рекурсивных s-местных функций. Там же было показано, что универсальная функция для класса $\mathfrak{F}^s_{\text{pr. r}}$ не может быть примитивно рекурсивной, а универсальная функция для класса всех общерекурсивных s-местных функций не может быть рекурсивной. Однако для совокупности $\mathfrak{F}^s_{\text{p. r}}$ частично рекурсивных функций положение иное.

Теорема 2. *Существует частично рекурсивная функция $T^{s+1}(x_0, x_1, \ldots, x_s)$, универсальная для совокупности всех s-местных частично рекурсивных функций.*

А именно, такой функцией заведомо является частичная функция

$$T^{s+1}(x_0, x_1, \ldots, x_s) = l(\mu_t (D^{s+2}(x_0, x_1, \ldots, x_s, t) = 0)), \quad (4)$$

где D^{s+2} — упомянутая выше общерекурсивная функция, универсальная для совокупности всех $(s+1)$-местных примитивно рекурсивных функций. Действительно, из формулы (4) непосредственно видно, что T^{s+1} — частично рекурсивная $(s+1)$-местная функция. С другой стороны, пусть $f(x_1, \ldots, x_s)$ — какая-нибудь s-местная частично рекурсивная функция. Представляем ее в форме Клини (1). В этой форме функция F примитивно рекурсивная и, следовательно, найдется такое число a, что

$$F(x_1, \ldots, x_s, t) = D^{s+2}(a, x_1, \ldots, x_s, t)$$

для всех значений $x_1, \ldots, x_s, t$. Подставляя

$$D^{s+2}(a_1, x_1, \ldots, x_s, t) \text{ вместо } F(x_1, \ldots, x_s, t)$$

в формулу (1), получаем

$$f(x_1, \ldots, x_s) = T^{s+1}(a, x_1, \ldots, x_s)$$

для всех значений $x_1, \ldots, x_s$. Следовательно, функция $T^{s+1}(x_0, x_1, \ldots, x_s)$ универсальная для совокупности $\mathfrak{F}^s_{\text{р. г}}$.

6.3. Доопределение функций. Построение нерекурсивного рекурсивно перечислимого множества. Частичная функция $g(x_1, \ldots, x_s)$ называется *расширением* частичной функции $f(x_1, \ldots, x_s)$, если в каждой точке, в которой функция f определена, функция g также определена и значение g в этой точке совпадает с значением f, то есть если график f содержится в графике g. Всюду определенное расширение частичной функции f называется *доопределением* f. Ясно, что каждая частичная функция имеет доопределение. Дело меняется, если ищутся не произвольные доопределения, а рекурсивные.

Теорема 1. *Никакая частично рекурсивная функция $U(x_0, x_1, \ldots, x_s)$, универсальная для совокупности всех s-местных частично рекурсивных функций, не может иметь общерекурсивных доопределений.*

Существует одноместная частично рекурсивная функция, принимающая лишь значения 0, 1 *и не имеющая рекурсивных доопределений.*

Введем частичную функцию

$$V(x) = \overline{\mathrm{sg}}\, U(x, x, \ldots, x). \tag{1}$$

Эта функция частично рекурсивная и принимает лишь значения 0 и 1. Допустим, что она имеет рекурсивное доопределение $W(x)$.

Рассматривая вместо $W(x)$ s-местную функцию

$$I^s_1(W(x_1), x_2, \ldots, x_s) = W(x_1)$$

и принимая во внимание универсальность функции U, видим, что для некоторого числа a и всех значений $x_1, \ldots, x_s$ имеет место равенство

$$W(x_1) = U(a, x_1, \ldots, x_s).$$

Полагая здесь $x_1 = \ldots = x_s = a$ и сравнивая результат с соотношением (1), получаем противоречивое равенство

$$W(a) = \overline{\mathrm{sg}}\, W(a).$$

Таким образом, функция $V(x)$ не имеет рекурсивных доопределений и второе утверждение теоремы 1 доказано. Если бы частичная функция $U(x_0, x_1, \ldots, x_s)$ имела рекурсивное доопределение $P(x_0, x_1, \ldots, x_s)$, то функция $\overline{\mathrm{sg}}\, P(x, x, \ldots, x)$ в силу (1) была бы рекурсивным доопределением $V(x)$, которое, как показано, не существует. Тем самым доказано и первое утверждение теоремы 1.

Теорема 2. *Существуют нерекурсивные рекурсивно перечислимые множества.*

Рассмотрим универсальную частично рекурсивную функцию $T^2(x_0, x_1)$, построенную в п. 6.2. Вводим функцию

$$E(x) = \overline{\mathrm{sg}}\, T^2(x, x).$$

Эта функция частично рекурсивная, принимающая лишь значения 0, 1 и не имеющая рекурсивных доопределений. Обозначим через $\boldsymbol{M}_0$ совокупность решений уравнения $E(x) = 0$. Согласно следствию 5 теоремы о графике (п. 6.1) множество $\boldsymbol{M}_0$ рекурсивно перечислимо. Докажем, что оно не рекурсивно. Пусть, напротив, $\boldsymbol{M}_0$ рекурсивно. Тогда характеристическая функция $\chi(x)$ множества $\boldsymbol{M}_0$ была бы рекурсивной. Но $\chi(x) = 0$ там, где $E(x) = 0$, и $\chi(x) = 1$ там, где $E(x) = 1$, то есть $\chi(x)$ есть рекурсивное доопределение функции $E(x)$, которая таких доопределений не имеет. Полученное противоречие доказывает теорему.

Следствие. *Существует примитивно рекурсивная одноместная функция, совокупность значений которой нерекурсивна.*

Действительно, построенное выше множество $\boldsymbol{M}_0$ рекурсивно перечислимое, но не рекурсивное. Поэтому оно непустое, а каждое непустое рекурсивно перечислимое множество есть совокупность значений подходящей примитивно рекурсивной функции $g(x)$.

Примитивно рекурсивные функции алгоритмически вычислимы. Следовательно, существует алгоритм, позволяющий вычислить значение $g(x)$ при любом x. В то же время нерекурсивность совокупности $\boldsymbol{M}_0$ всех значений

функции $g(x)$ означает, что не существует алгоритм, позволяющий для произвольного числа a узнать, входит a в $\boldsymbol{M}_0$ или нет, то есть разрешимо или нет уравнение $g(x) = a$.

Итак, *совокупность значений примитивно рекурсивной функции может не быть рекурсивным множеством*. Поэтому известный интерес представляет тот факт, что *совокупность значений монотонно растущей общерекурсивной функции является рекурсивным множеством* (п. 4.1, теорема 3).

Теорема 3. *Каждое бесконечное рекурсивно перечислимое множество* $\boldsymbol{A}$ *есть совокупность значений подходящей общерекурсивной функции* $g(t)$, *принимающей для различных значений* t *различные значения.*

Иначе говоря, *для каждого бесконечного рекурсивно перечислимого множества* $\boldsymbol{A}$ *существует общерекурсивная функция* $g(t)$, *отображающая взаимно однозначно натуральный ряд на* $\boldsymbol{A}$.

По условию $\boldsymbol{A}$ есть совокупность значений некоторой примитивно рекурсивной функции $u(t)$. Искомую функцию $g(t)$ строим следующим образом. Полагаем по определению $g(0) = u(0)$. Пусть значения $g(0), g(1), \ldots, g(n)$ уже определены. Тогда берем наименьшее t такое, что $u(t) \notin \{g(0), g(1), \ldots, g(n)\}$ и полагаем $g(n+1) = u(t)$. Существование такого значения для t вытекает из бесконечности множества $\boldsymbol{A}$. Поэтому функция $g(n)$ всюду определенная. Множество всех ее значений, очевидно, совпадает с множеством значений функции u. Для формального доказательства рекурсивности g выразим ее через u с помощью основных операций. Положим

$$v(n) = \mu_t \left(\prod_{i=0}^{n} |u(t) - u(i)| \neq 0 \right). \tag{2}$$

Число $v(n)$ есть наименьшее значение t, для которого

$$u(t) \notin \{u(0), u(1), \ldots, u(n)\}.$$

Сравнивая это с определением функции $g(x)$, видим, что

$$g(1) = u(v(0)), \quad g(2) = u(v(v(0))), \ldots$$

Поэтому, вводя еще функцию $w(x)$ посредством рекурсии

$$w(0) = 0, \quad w(n+1) = v(w(n)), \tag{3}$$

будем иметь окончательно: $g(x) = u(w(x))$ и, значит, в силу (2), (3) функция $g(x)$ общерекурсивная.

В заключение докажем еще одну простую теорему о существовании частично рекурсивных продолжений некоторых функций.

Теорема 4. *Если функции $f_i(x_1, \ldots, x_s)$ и $g_i(x_1, \ldots, x_s)$ $(i = 1, \ldots, n)$ частично рекурсивные и частичная функция $h(x_1, \ldots, x_s)$ удовлетворяет условиям*

$$h(x_1, \ldots, x_s) = \begin{cases} f_1(x_1, \ldots, x_s), & \text{если} \quad g_1(x_1, \ldots, x_s) = 0, \\ f_2(x_1, \ldots, x_s), & \text{если} \quad g_2(x_1, \ldots, x_s) = 0, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ f_n(x_1, \ldots, x_s), & \text{если} \quad g_n(x_1, \ldots, x_s) = 0, \\ \text{неопределенная для остальных } x_1, \ldots, x_s, & \end{cases}$$

то h — также частично рекурсивная функция.

В силу теоремы о графике достаточно показать, что график $\boldsymbol{H}$ функции h рекурсивно перечислимый. Обозначим через $\boldsymbol{A}_i$ совокупность тех последовательностей $\langle x_1, \ldots, x_s, y\rangle$, которые удовлетворяют уравнениям

$$|f_i(x_1, \ldots, x_s) - y| + g_i(x_1, \ldots, x_s) = 0, \; i = 1, \ldots, n. \quad (4)$$

Ясно, что $\boldsymbol{H}$ есть объединение совокупностей $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_n$. Левая часть уравнения (4) есть частично рекурсивная функция от переменных $x_1, \ldots, x_s, y$. Поэтому в силу следствия 5 теоремы 1 (п. 6.1) множество $\boldsymbol{A}_i$ рекурсивно перечислимое. Из рекурсивной перечислимости множеств $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_n$ вытекает рекурсивная перечислимость их объединения $\boldsymbol{H}$, что и требовалось.

Следствие 1. *Если область определенности $\boldsymbol{M}$ частично рекурсивной функции $f(x_1, \ldots, x_s)$ рекурсивная, то f имеет рекурсивное доопределение.*

По условию характеристическая функция χ множества $\boldsymbol{M}$ общерекурсивная. Рассмотрим функцию g, заданную схемой

$$g(x_1, \ldots, x_s) = \begin{cases} f(x_1, \ldots, x_s), & \text{если} \quad \chi(x_1, \ldots, x_s) = 0, \\ 0, & \text{если} \quad |\chi(x_1, \ldots, x_s) - 1| = 0. \end{cases}$$

Эта функция всюду определенная и согласно предыдущей теореме частично рекурсивная. В то же время она является расширением функции f.

С л е д с т в и е 2. *Если частично рекурсивная функция $U(x_0, x_1, \ldots, x_s)$ является универсальной для совокупности всех s-местных частично рекурсивных функций, то область определенности* $\boldsymbol{M}$ *этой функции есть нерекурсивное рекурсивно перечислимое множество.*

Действительно, множество $\boldsymbol{M}$, как область определенности частично рекурсивной функции, рекурсивно перечислимое. Если бы $\boldsymbol{M}$ оказалось рекурсивным, то согласно следствию 1 функция U была бы рекурсивно доопределимой в противоречие с теоремой 1.

6.4. Исследование представления Клини. Ввиду важности нормального представления Клини (п. 6.2) представляет интерес более детальное исследование этого представления, произведенное Сколемом и А. А. Марковым. Прежде всего возникает вопрос: нельзя ли в представлении Клини обойтись без внешней функции l? Полный ответ на этот вопрос дает

Т е о р е м а 1 (С к о л е м [93]). *Для того чтобы функция $f(x_1, \ldots, x_s)$ была представима в виде*

$$f(x_1, \ldots, x_s) = \mu_t(Q(x_1, \ldots, x_s, t) = 0), \tag{1}$$

где $Q(x_1, \ldots, x_s, t)$ — подходящая примитивно рекурсивная функция, необходимо и достаточно, чтобы график функции f был примитивно рекурсивным.

Достаточность очевидна, так как примитивная рекурсивность графика f означает, что примитивно рекурсивна его характеристическая функция, равная $\chi(x_1, \ldots, x_s, y)$. Но тогда очевидное равенство

$$f(x_1, \ldots, x_s) = \mu_y(\chi(x_1, \ldots, x_s, y) = 0)$$

дает сразу же искомое представление f в виде (1).

Н е о б х о д и м о с т ь. Пусть f имеет вид (1). Вводим функцию

$$G(x_1, \ldots, x_s, x) = \left|1 - \sum_{n=0}^{x} \overline{\mathrm{sg}} \prod_{i=0}^{n} Q(x_1, \ldots, x_s, i)\right|. \tag{2}$$

В силу результатов п. 3.2 функция G примитивно рекурсивна. Кроме того, из формулы (2) видно, что $y = f(x_1, \ldots, x_s)$ тогда и только тогда, когда y есть наименьший корень уравнения $G(x_1, \ldots, x_s, y) = 0$. Следовательно,

$$f(x_1, \ldots, x_s) = \mu_t(G(x_1, \ldots, x_s, t) = 0). \tag{3}$$

Представление (3) имеет тот же вид, что и представление (1). Особенность состоит в том, что теперь число $f(x_1, \ldots, x_s) = t$ является

единственным решением уравнения $G(x_1, \ldots, x_s, t) = 0$. Но в таком случае функция $\operatorname{sg} G(x_1, \ldots, x_s, y)$ будет характеристической функцией графика f и, значит, этот график примитивно рекурсивен.

Нам остается показать, что в форме Сколема (1) представима не каждая частично рекурсивная функция. Согласно теореме 1 для этого достаточно построить общерекурсивную функцию, график которой не является примитивно рекурсивным.

В качестве исходного материала берем построенную в п. 5.2 общерекурсивную функцию $D(x_0, x_1)$, универсальную для совокупности всех одноместных примитивно рекурсивных функций. В качестве искомой функции можно взять функцию

$$V(x) = \overline{\operatorname{sg}}\, D(x, x).$$

Действительно, эта функция не может быть примитивно рекурсивной, так как в противном случае для подходящего числа n мы имели бы тождество $V(x) = D(n, x)$, из которого получили бы соотношение

$$D(n, n) = V(n) = \overline{\operatorname{sg}}\, D(n, n),$$

противоречащее определению функции $\overline{\operatorname{sg}}$. Характеристической функцией графика V является функция

$$\chi(x, y) = \operatorname{sg} |y - V(x)|.$$

Если бы эта функция была примитивно рекурсивной, то такой же была бы и одноместная функция

$$\chi(x, 0) = V(x),$$

что противоречит доказанному выше. Таким образом, общерекурсивная функция $V(x)$ не может быть представлена в форме Сколема (1).

Чтобы получить характеристику тех функций L, которые могут играть роль «внешней» функции в теореме Клини о специальной форме (п. 6.2), введем, следуя А. А. Маркову [54], понятие функции большого размаха. Именно, будем говорить, что функция $L(x)$ имеет *большой размах*, если для каждого натурального a уравнение $L(x) = a$ имеет бесконечно много решений. Например, функциями большого размаха являются $\mathrm{ex}_n\, x$, $x - [\sqrt{x}]^2$ и т. д.

Лемма 1. *Для каждой примитивно рекурсивной функции большого размаха $L(x)$ существуют примитивно рекурсивная функция $R(x)$ и общерекурсивная функция $C(x, y)$, связанные тождествами*

$$L(C(x, y)) = x, \quad R(C(x, y)) = y, \qquad C(L(x), R(x)) = x. \tag{4}$$

Иначе говоря, каждая функция большого размаха связана с функцией, осуществляющей простую однозначную нумерацию пар натуральных чисел (ср. п. 3.3)

По условию для каждого значения x уравнение $L(y) = L(x)$ имеет бесконечно много решений для y. Пусть $y_0, y_1, y_2, \ldots$ — эти решения, расположенные в порядке возрастания. Среди них должно встретиться и само x: $x = y_n$. Номер n будет функцией от x: $n = R(x)$. Таким образом, $y_{R(x)} = x$. Из очевидной формулы

$$R(x) = \sum_{i=0}^{x} \overline{\operatorname{sg}} \, | L(x) - L(i) | \dot{-} 1 \qquad (5)$$

следует, что функция $R(x)$ примитивно рекурсивна.

Покажем теперь, что в последовательности

$$\langle L(0), R(0) \rangle, \langle L(1), R(1) \rangle, \ldots, \langle L(n), R(n) \rangle, \ldots \qquad (6)$$

встречается каждая пара натуральных чисел и только один раз.

Действительно, из определения функции $R(x)$ выводим, что номером произвольной пары чисел $\langle a, b \rangle$ в последовательности (6) служит решение номер b уравнения $L(x) = a$.

Теперь легко устанавливается, что функция

$$C(x, y) = \mu_t (| x - L(t) | + | y - R(t) | = 0)$$

связана с функциями L и R тождествами (4).

Теорема 2 (Марков [54]). *Если $L(x)$ — примитивно рекурсивная функция большого размаха, то для каждого $s > 0$ каждая s-местная частично рекурсивная функция f может быть представлена в виде*

$$f(x_1, \ldots, x_s) = L(\mu_t (F(x_1, \ldots, x_s, t) = 0)), \qquad (7)$$

где F — подходящая (зависящая от f) примитивно рекурсивная функция.

Если для какого-нибудь числа $s > 0$ и какой-нибудь примитивно рекурсивной функции $L(x)$ каждая s-местная общерекурсивная функция f представима в клиниевской форме (7), *то $L(x)$ — функция большого размаха.*

Докажем первое утверждение. Согласно лемме 1 существуют примитивно рекурсивная функция $R(x)$ и общерекурсивная функция $C(x, y)$, составляющие вместе с функцией $L(x)$ тройку нумерующих функций. В таком случае согласно теореме Клини о нормальной форме (п. 6.2, замечание к теореме 1), каждая s-местная частично рекурсивная функция f может быть представлена в виде (7).

Второе утверждение будем доказывать по способу от противного, а именно, покажем, что для каждой примитивно рекурсивной функции $L(x)$, не имеющей большого размаха, существует общерекурсивная функция f, которую нельзя представить в виде (7). По условию должно существовать такое число a, для которого уравнение $L(x) = a$ имеет не более конечного числа решений. Если это уравнение совсем не имеет решений, то в виде (7) нельзя представить даже функцию $f(x) = x$. Допустим, что уравнение $L(x) = a$ имеет решения $b_1, \ldots, b_m$ и пусть $f(x)$ — какая-нибудь функция, имеющая вид (7). Как уже сказано, мы можем считать при этом, что для каждого x уравнение $F(x, t) = 0$ имеет не более одного решения t. Мы теперь хотим доказать, что при сделанных

предположениях совокупность решений уравнения $f(x)=a$ будет непременно примитивно рекурсивным множеством и, значит, никакая функция $f(x)$, у которой множество решений указанного уравнения не примитивно рекурсивно, не может быть представлена в виде (7). Но такую функцию построить очень легко. Например, если $a=0$, то требуемым свойством обладает функция $V(x)=\overline{\mathrm{sg}}\, D(x, x)$. Если же $a \neq 0$, то требуемым свойством обладает функция $aV(x)$.

Итак, рассмотрим решения уравнения $f(x)=a$. Их совокупность есть объединение решений уравнений

$$\mu_t (F(x, t)=0)=b_i \quad (i=1, \ldots, m),$$

то есть уравнений $F(x, b_i)=0$. Поскольку функция $F(x, t)$ примитивно рекурсивная, то совокупность решений каждого уравнения $F(x, b_i)=0$ примитивно рекурсивна (п. 4.1), а потому примитивно рекурсивной будет и совокупность решений уравнения $f(x)=a$, что и требовалось.

Дополнения, примеры и задачи

1. Если график всюду определенной функции рекурсивный, то функция также рекурсивная. Существует не примитивно рекурсивная всюду определенная функция, график которой примитивно рекурсивен (см. функцию $\mu_t (Q=0)$ в формуле (1) из п. 6.4).

2. Обозначим через $\min_t (f(x, t)=0)$ наименьшее из решений t уравнения $f(x, t)=0$, если решения существуют. Если же решений нет, то значение указанного выражения будем считать неопределенным. Оператор $\min_t$ отличается от оператора минимизации μ_t. Например,

$$\mu_t (t-(x+1)=0)=\text{неопр.},$$

$$\min_t (t-(x+1)=0)=x+1.$$

Построить такую частично рекурсивную функцию $f(x, t)$, для которой функция

$$g(x)=\min_t (f(x, t)=0)$$

не является частично рекурсивной.

3. Частичная функция $f(x_1, \ldots, x_n)$ тогда и только тогда представима в форме

$$f(x_1, \ldots, x_n)=\mu_t (g(x_1, \ldots, x_n, t)=0), \tag{a}$$

где g — подходящая общерекурсивная функция, когда график f рекурсивный. (Для доказательства необходимости заменить g в (a) функцией g^*, для которой уравнение $g^*(x_1, \ldots, x_n, t)=0$ имеет не более одного решения t, после чего будем иметь $f(\mathfrak{X})=y \Longleftrightarrow g^*(\mathfrak{X}, y)=0$, где положено $\mathfrak{X}=\langle x_1, \ldots, x_n \rangle$.)

Частичная характеристическая функция множества M тогда и только тогда представима в виде (a), когда M рекурсивно.

4. *Элементарными* по Кальмару [38] функциями называются всюду определенные функции от произвольного числа аргу-

ментов, которые можно получить из функций 1, I^n_m, $x+y$, $x \dot{-} y$ с помощью конечного числа операций подстановки, суммирования и мультиплицирования (см. п. 3.1).

Ясно, что все элементарные по Кальмару функции примитивно рекурсивны. Обратное неверно. Пусть функция $\xi(a, n)$ определяется рекурсией $\xi(a, 0) = 1$, $\xi(a, n+1) = a^{\xi(a, n)}$. Показать, что каждая элементарная по Кальмару функция $f(x_1, \ldots, x_n)$ удовлетворяет неравенству

$$f(x_1, \ldots, x_n) \leqslant \xi(2+x_1+\ldots+x_n, c), \tag{b}$$

где c — константа (зависящая от f). Отсюда, в частности, следует, что функция $\xi(x, y)$ не элементарна (Берецки [7]).

Верно ли, что каждая примитивно рекурсивная функция f, удовлетворяющая неравенству вида (b), элементарна?

5. Говорят, что функция $f(x_1, \ldots, x_n)$ получается явной трансформацией из функции $g(x_1, \ldots, x_m)$, если

$$f(x_1, \ldots, x_n) = g(u_1, \ldots, u_m),$$

где каждое u_i — или константа, или какая-то переменная x_j. Функция $f(x_1, \ldots, x_n, x)$ называется возникающей из функций g, h, α ограниченной рекурсией, если

$$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n),$$
$$f(x_1, \ldots, x_n, x+1) = h(x_1, \ldots, x_n, x, f(x_1, \ldots, x_n, x)),$$
$$f(x_1, \ldots, x_n, x) \leqslant \alpha(x_1, \ldots, x_n, x).$$

Показать, что элементарные по Кальмару функции — это те и только те функции, которые можно получить из функций $x+1$, x^y конечным числом явных трансформаций и ограниченных рекурсий (А. Гжегорчик [21]).

6. Функцией, элементарной по Сколему [95], называется функция, которую можно получить из функций 1, I^n_m, $x+y$, $x \dot{-} y$ операциями подстановки и суммирования из п. 3.1. Легко видеть, что функции xy, $[x/y]$, $l(x)$, $r(x)$, $\Gamma(x, y)$ элементарны по Сколему. Показать, что каждая элементарная по Сколему функция $f(x_1, \ldots, x_n)$ удовлетворяет неравенству вида

$$f(x_1, \ldots, x_n) \leqslant (2+x_1+\ldots+x_n)^c,$$

где c — константа, зависящая от f. В частности, элементарные по Кальмару функции 2^x, 2^{2^x} не элементарны по Сколему.

7. Класс $\mathfrak{K}$ некоторых общерекурсивных функций назовем достаточным, если для любого n каждая непустая рекурсивно перечислимая совокупность n-ок чисел является совокупностью n-ок вида

$$\langle \alpha_1(t), \ldots, \alpha_n(t) \rangle \qquad (t = 0, 1, \ldots),$$

где $\alpha_1(t), \ldots, \alpha_n(t)$ — подходящие функции класса $\mathfrak{K}$. Например, заведомо достаточным является класс всех примитивно рекурсивных функций. Возникает задача о нахождении других, по возможности более простых и узких достаточных классов.

Анализируя доказательство теоремы о графике частично рекурсивной функции (п. 6.1), показать, что класс элементарных по Сколему функций является достаточным.

8. В каждом бесконечном рекурсивно перечислимом множестве содержится бесконечное рекурсивное подмножество. Существуют бесконечные рекурсивно перечислимые множества, не содержащие бесконечных примитивно рекурсивных множеств (см. следующую задачу).

9. Пусть $\mathfrak{A}$ — совокупность значений функции Аккермана $A(x)$ из п. 5.3. Эта совокупность рекурсивная. Показать, что $\mathfrak{A}$ не может быть совокупностью значений никакой примитивно рекурсивной функции $f(x)$, принимающей различные значения при различных значениях аргумента. (Допустим противное. Так как $A(x)$ монотонно возрастает, то на отрезке натурального ряда $0, 1, \ldots, A(x)$ содержится ровно $x+1$ чисел множества $\mathfrak{A}$. Берем числа $f(0), f(1), \ldots, f(x)$. Если они все лежат на указанном отрезке, то самое большое из них совпадает с $A(x)$. Если нет, то самое большое из них больше $A(x)$. Поэтому, полагая $g(x) = \sum_{i=0}^{x} f(i)$, будем иметь $A(x) \leqslant g(x)$ для всех значений x. Ввиду примитивной рекурсивности функции $g(x)$ указанное неравенство противоречит основному свойству функции Аккермана — расти быстрее всякой примитивно рекурсивной функции.)

10. Пусть F, G—частично рекурсивные одноместные функции, A — объединение областей определенности этих функций. Показать, что определенная ниже функция H имеет своей областью определенности множество A и в каждой точке $x \in A$ значение $H(x)$ или совпадает с $F(x)$, или совпадает с $G(x)$.

Пусть области определенности функций F и G суть множества значений примитивно рекурсивных функций $f(x)$ и соответственно $g(x)$. Полагаем

$$h(x) = \mu_t\,(f(t) = x \vee (f(t) \neq x \,\&\, g(t) = x)),$$

$$H(x) = \begin{cases} F(x), & \text{если } f(h(x)) = x, \\ G(x), & \text{если } g(h(x)) = x. \end{cases}$$

Аналогичное построение проходит и для функций от большего числа переменных (см. п. 7.3).

11. Основываясь на глубокой теореме (см. § 16) о представимости рекурсивно перечислимых совокупностей n-ок чисел в экспоненциально диофантовой форме, Д е в и с [25] показал, что существуют функции $P(x)$, $g(x, y, z)$, $h(x, y, z)$, возникающие конечным числом подстановок из функций o, I^n_m, xy, 2^x, $l(x)$, $r(x)$, такие, что каждая одноместная частично рекурсивная функция $f(x)$ представима в форме

$$f(x) = P(\mu_t\,(g(n, x, t) = h(n, x, t))).$$

12. Бинарное отношение $\prec$, определенное на натуральном ряде, назовем простым конструктивным расположением натураль-

ных чисел, если существует общерекурсивная функция $\sigma(x)$, значение которой $\sigma(n)$ есть «ближайшее за n большее число» в смысле порядка $<$ и если отношение $<$ упорядочивает натуральные числа в простую последовательность, то есть если все натуральные числа содержатся в последовательности

$$a < \sigma(a) < \sigma(\sigma(a)) < \ldots < \sigma^k(a) < \ldots$$

Показать, что каждая общерекурсивная функция $f(x)$ представима в форме $f(x) = g(2^x)$, где g определяется схемой

$$g(a) = a_0, \qquad (\alpha)$$
$$g(x) = h(x,\ g(\alpha(x))) \quad (x \neq a).$$

Здесь a — наименьшее число в подходящем конструктивном простом порядке $<$, $\alpha(x)$ и $h(x, y)$ — примитивно рекурсивные функции, причем $\alpha(x) < x$ для $x \neq a$ (см. Лю [49], Майхил [58]).

13. Показать, что существуют общерекурсивные одноместные функции, не представимые в виде $\varphi(g(x))$, где φ примитивно рекурсивная, а g определяется схемой (α) из предыдущей задачи (Проблема Раутледжа. Решение см. Лю [48]).

ГЛАВА IV

НУМЕРОВАННЫЕ СОВОКУПНОСТИ

Для того чтобы понятия рекурсивности и рекурсивной перечислимости перенести из области натуральных чисел в область более сложных объектов — n-ок чисел,— мы предварительно занумеровали все n-ки натуральными числами. В данной главе метод нумерации будет рассмотрен в общем виде. Он позволяет глубже вскрыть природу алгоритмических процессов и прямым путем приводит к решению многих интересных проблем. Впервые метод нумерации для решения фундаментальных вопросов математической логики был применен К. Геделем, вследствие чего многие важные нумерации обычно называются нумерациями Геделя. При более детальном изучении конкретных нумераций Геделя целесообразно, однако, ввести особые названия для некоторых из этих нумераций, что ниже и сделано.

§ 7. НУМЕРАЦИИ СОВОКУПНОСТЕЙ МНОЖЕСТВ И ФУНКЦИЙ

Изучение нумераций произвольных совокупностей объектов мы начнем с подробного изучения специальных нумераций совокупностей всех одноместных частично рекурсивных функций и всех рекурсивно перечислимых множеств, которые будем называть нумерациями Клини и Поста.

7.1. Универсальные функции Клини. При детальном изучении свойств частично рекурсивных функций удобнее пользоваться не универсальными функциями $T^{n+1}(x_0, x_1, \ldots, x_n)$, построенными в п. 6.2, а особыми функциями Клини, к определению которых мы и перейдем.

Введем стандартное обозначение

$$[x, y] = c(l(x), c(r(x), y)), \tag{1}$$

где $c(x, y)$ — канторовский номер пары $\langle x, y\rangle$, а $l(x)$ и $r(x)$ — левое и правое число пары с номером x (п. 3.3). Из тождеств

$$c(l(x), r(x)) = x, \quad l(c(x, y)) = x, \quad r(c(x, y)) = y$$

и формулы (1) непосредственно следует, что если $[x, y] = n$, то

$$\begin{aligned} x &= c(l(n), l(r(n))), \\ y &= r(r(n)). \end{aligned} \tag{2}$$

Поэтому, вводя обозначения

$$[x]_{21} = c(l(x), l(r(x))), \qquad [x]_{22} = r(r(x)),$$

будем иметь тождества

$$[[x]_{21}, [x]_{22}] = x, \quad [[x, y]]_{21} = x, \quad [[x, y]]_{22} = y,$$

показывающие, что функция $[x, y]$, так же как и функция Кантора $c(x, y)$, осуществляет взаимно однозначную нумерацию пар натуральных чисел.

Положив еще по определению

$$[x_1, x_2, \ldots, x_s] = [\ldots[[x_1, x_2], x_3], \ldots, x_s] \quad (s = 3, 4, \ldots), \tag{3}$$

получим взаимно однозначную нумерацию s-ок чисел. Решая уравнение

$$[x_1, \ldots, x_s] = x \tag{4}$$

относительно $x_1, \ldots, x_s$, будем иметь

$$\begin{aligned} x_1 &= [[\ldots[x]_{21}]_{21} \ldots]_{21}, \\ x_2 &= [[\ldots [[x]_{21}]_{21} \ldots]_{21}]_{22}, \\ &\ldots\ldots\ldots\ldots\ldots \\ x_s &= [x]_{22}. \end{aligned} \tag{5}$$

Правые части этих равенств обозначим соответственно через $[x]_{s1}, \ldots, [x]_{ss}$. Тогда из (4), (5) получим тождества

$$[[x]_{s1}, \ldots, [x]_{ss}] = x, \quad [[x_1, \ldots, x_s]]_{si} = x_i.$$

Отметим еще тождество

$$[x_1, \ldots, x_m, x_{m+1}, \ldots, x_s] = [[x_1, \ldots, x_m], x_{m+1}, \ldots, x_s], \tag{6}$$

непосредственно вытекающее из (3).

Функции Кантора $c(x, y)$, $l(x)$, $r(x)$ — примитивно рекурсивные функции, выражающиеся через свои аргументы при помощи элементарных арифметических операций $+$, $\times$, $\dot{-}$, $[\sqrt{\ }]$, $[\frac{}{2}]$. Формулы (1), (2) показывают, что такими же будут и функции $[x, y]$, $[x]_{ij}$.

Теперь мы вместо универсальных функций $T^{s+1}(x_0, x_1, \ldots, x_s)$, построенных в п. 6.2, введем следующие функции Клини:

$$\begin{gathered} K^2(x_0, x_1) = T^2(l(x_0), c(r(x_0), x_1)), \\ K^{s+1}(x_0, x_1, \ldots, x_s) = K^s([x_0, x_1], x_2, \ldots, x_s) \\ (s = 2, 3, \ldots). \end{gathered} \tag{7}$$

Из тождества (6) вытекает важное соотношение

$$K^{s+t+1}(x_0, \ldots, x_{s+t}) = K^{t+1}([x_0, \ldots, x_s], x_{s+1}, \ldots, x_{s+t}), \tag{8}$$

связывающее функции Клини, имеющие разные числа аргументов.

Лемма 1. *Функция Клини $K^s(x_1, \ldots, x_s)$ связана с универсальной функцией $T^{s+1}(x_0, x_1, \ldots, x_s)$ тождеством*

$$K^s(c(x_0, x_1), x_2, \ldots, x_s) = T^{s+1}(x_0, x_1, \ldots, x_s). \tag{9}$$

Прежде всего заметим, что нумерации $c(x, y)$ и $[x, y]$ удовлетворяют своеобразному закону ассоциативности

$$c(x_0, c(x_1, x_2)) = [c(x_0, x_1), x_2], \tag{10}$$

легко проверяемому с помощью тождеств, связывающих функции $l(x)$, $r(x)$, $[x, y]$. Далее имеем

$$K^2(c(x_0, x_1), x_2) = T^2(x_0, c(x_1, x_2)) = T^3(x_0, x_1, x_2).$$

Аналогично,

$$\begin{aligned} K^3(c(x_0, x_1), x_2, x_3) &= K^2([c(x_0, x_1), x_2], x_3) = \\ &= K^2(c(x_0, c(x_1, x_2)), x_3) = T^3(x_0, c(x_1, x_2), x_3) = \\ &= T^4(x_0, x_1, x_2, x_3) \end{aligned}$$

и т. д.

Теорема 1. *Функция Клини $\boldsymbol{K}^{n+1}(x_0, x_1, \ldots, x_n)$ является частично рекурсивной функцией, универсальной для совокупности всех n-местных частично рекурсивных функций от $x, \ldots, x_n$.*

Пусть $f(x_1, \ldots, x_n)$ — какая-нибудь частично рекурсивная функция. Из универсальности функции T^{n+2} следует, что при некотором фиксированном a будем иметь тождество

$$0 \cdot x + f(x_1, \ldots, x_n) = T^{n+2}(a, x, x_1, \ldots, x_n) = \\ = \boldsymbol{K}^{n+1}(\boldsymbol{c}(a, x), x_1, \ldots, x_n).$$

Полагая в нем $x = 0$ и обозначая число $\boldsymbol{c}(a, 0)$ через b, получим тождество

$$f(x_1, \ldots, x_n) = \boldsymbol{K}^{n+1}(b, x_1, \ldots, x_n),$$

которое и доказывает, что функция $\boldsymbol{K}^{n+1}(x_0, x_1, \ldots, x_n)$ универсальна для частично рекурсивных функций от переменных $x_1, \ldots, x_n$.

7.2. Нумерация Клини. Ставя каждому натуральному числу n в соответствие одноместную функцию $\boldsymbol{K}^2(n, x)$, мы получим отображение множества всех натуральных чисел $\boldsymbol{N}$ на совокупность всех одноместных частично рекурсивных функций. Это отображение мы будем далее называть *нумерацией Клини* и обозначать через $\varkappa$. Там, где не возникнет опасность наложения обозначений, мы будем вместо $\boldsymbol{K}^{s+1}(x_0, x_1, \ldots, x_s)$ писать просто $\boldsymbol{K}(x_0, x_1, \ldots, x_s)$. Число n называется *клиниевским номером* функции $\boldsymbol{K}(n, x)$, которая будет обозначаться также через $\varkappa n$ и иногда через K_n.

Теорема 1. *Для каждой частично рекурсивной функции $f(x)$ существует такое фиксированное число a, что все числа последовательности*

$$[a, 0], [a, 1], \ldots, [a, m], \ldots \tag{1}$$

будут клиниевскими номерами функции $f(x)$.

По условию $f(x) = \boldsymbol{K}(n, x)$. Так как $\boldsymbol{K}(x_0, y, x)$ — универсальная функция, то найдется такое число a, что при любых значениях переменных y, x

$$0 \cdot y + f(x) = \boldsymbol{K}(a, y, x) = \boldsymbol{K}([a, y], x).$$

Поэтому последовательность (1) действительно состоит из номеров $f(x)$.

Рассмотрим следующую задачу: *даны клиниевские номера функций $f(x)$ и $g(x)$. Требуется найти клиниевские номера их суммы и суперпозиции.*

Рассуждаем так. Берем следующие функции от переменных u, v, x:

$$\boldsymbol{K}(u, x)+\boldsymbol{K}(v, x), \ \boldsymbol{K}(u, \boldsymbol{K}(v, x)).$$

Так как $\boldsymbol{K}(x_0, u, v, x)$ — универсальная функция, то найдутся такие числа a, b, что тождественно относительно u, v, x

$$\boldsymbol{K}(u, x)+\boldsymbol{K}(v, x)=\boldsymbol{K}(a, u, v, x)=\boldsymbol{K}([a, u, v], x),$$

$$\boldsymbol{K}(u, \boldsymbol{K}(v, x))=\boldsymbol{K}(b, u, v, x)=\boldsymbol{K}([b, u, v], x).$$

Следовательно, если m, n — номера функций f, g, то числа $[a, m, n]$, $[b, m, n]$ заведомо будут номерами $f+g$ и $f(g(x))$.

Аналогичные утверждения имеют место для μ-операции, операции примитивной рекурсии и других операций.

Несмотря на простоту доказательства, следующая теорема играет значительную роль во многих рассуждениях.

Теорема 2 (теорема Клини о неподвижной точке). *Для каждой частично рекурсивной функции $h(x_1, \ldots, x_n, x_{n+1})$ существует такая примитивно рекурсивная функция $g(x_1, \ldots, x_n)$, что*

$$\varkappa h(x_1, \ldots, x_n, g(x_1, \ldots, x_n))=\varkappa g(x_1, \ldots, x_n). \qquad (2)$$

В частности, *для каждой частично рекурсивной функции $h(x)$ существует такое число a («неподвижная точка» отображения h), что*

$$\varkappa h(a)=\varkappa a. \qquad (3)$$

Докажем первое утверждение. Рассмотрим вспомогательную функцию $\boldsymbol{K}(h(x_1, \ldots, x_n, [y, y, x_1, \ldots, x_n]), t)$. Из универсальности функции $\boldsymbol{K}(x_0, y, x_1, \ldots, x_n, t)$ следует, что при подходящем числе a имеем тождество

$$\boldsymbol{K}([a, y, x_1, \ldots, x_n], t)=$$
$$=\boldsymbol{K}(h(x_1, \ldots, x_n, [y, y, x_1, \ldots, x_n]), t).$$

Полагая здесь $y=a$, $[a, a, x_1, \ldots, x_n]=g(x_1, \ldots \ldots, x_n)$, получим искомое тождество (2). Соотношение (3)

можно рассматривать как частный случай соотношения (2) при $n = 0$.

В связи с нумерацией Клини естественно возникает вопрос о том, какие свойства частично рекурсивных функций можно эффективно распознать по номерам функций. Например, естественно спросить, существует ли алгоритм, позволяющий для произвольного n узнать, является ли функция с номером n нигде не определенной, тождественно равной нулю, примитивно рекурсивной и т. п. В точных терминах этот вопрос можно изложить следующим образом: является ли рекурсивной совокупность клиниевских номеров нигде не определенной функции, тождественного нуля, всех примитивно рекурсивных функций? Нижеследующая теорема дает отрицательный ответ на все эти вопросы: никакое нетривиальное (то есть присущее некоторым, но не всем функциям) свойство одноместных частично рекурсивных функций не может быть эффективно распознаваемым по клиниевским номерам функций. Иными словами, справедлива следующая

Теорема 3 (Райс [81]). *Если $\mathfrak{S}$ — непустое семейство одноместных частично рекурсивных функций, отличное от совокупности всех таких функций, то множество клиниевских номеров функций, принадлежащих $\mathfrak{S}$, не может быть рекурсивным.*

Пусть, напротив, множество $\boldsymbol{A}$ клиниевских номеров всех функций из $\mathfrak{S}$ рекурсивно. Тогда рекурсивным будет и множество $\boldsymbol{A}'$ клиниевских номеров всех функций, не принадлежащих $\mathfrak{S}$ ($\boldsymbol{A}'$ — дополнение $\boldsymbol{A}$). По условию и в $\mathfrak{S}$ и вне $\mathfrak{S}$ функции есть. Поэтому $\boldsymbol{A}$ и $\boldsymbol{A}'$ не пусты. Выберем в $\boldsymbol{A}$ и $\boldsymbol{A}'$ по одному числу a, b и определим следующую функцию

$$g(x) = \begin{cases} a, & x \in \boldsymbol{A}', \\ b, & x \in \boldsymbol{A}. \end{cases} \tag{4}$$

Согласно теореме 4 (п. 6.3) функция $g(x)$ общерекурсивна. Применяя теорему о неподвижной точке, приходим к выводу, что должно существовать число n, для которого

$$\varkappa g(n) = \varkappa n. \tag{5}$$

Какое из множеств $\boldsymbol{A}$, $\boldsymbol{A}'$ содержит n? Если $n \in \boldsymbol{A}$, то при помощи (5) получаем

$$n \in \boldsymbol{A} \Rightarrow \varkappa n \in \mathfrak{S} \Rightarrow \varkappa g(n) \in \mathfrak{S}. \qquad (6)$$

С другой стороны, принимая во внимание (4), имеем

$$n \in \boldsymbol{A} \Rightarrow g(n) = b \Rightarrow \varkappa g(n) = \varkappa b \in \mathfrak{S}'. \qquad (7)$$

Заключения (6) и (7) противоречат друг другу и потому n не может принадлежать $\boldsymbol{A}$. Меняя ролями множества $\boldsymbol{A}$ и $\boldsymbol{A}'$, получим, что n не может принадлежать $\boldsymbol{A}'$. Мы получили противоречие, так как каждое число должно входить либо в $\boldsymbol{A}$, либо в его дополнение $\boldsymbol{A}'$. Теорема 3 доказана.

Итак, совокупность всех клиниевских номеров всех функций любого нетривиального семейства функций $\mathfrak{S}$ не может быть рекурсивным множеством. Однако для некоторых частных семейств $\mathfrak{S}$ совокупность $\boldsymbol{A}$ будет рекурсивно перечислимой. Мы сейчас не будем заниматься разысканием всех семейств $\mathfrak{S}$, обладающих указанным свойством (см. задачи 9, 10 в конце этого §), а ограничимся лишь следующим очевидным утверждением.

Теорема 4. *Множество $\boldsymbol{A}$ клиниевских номеров всех одноместных частично рекурсивных функций, принимающих в данной точке $x = a$ данное значение b, рекурсивно перечислимо. Множество $\boldsymbol{B}$ всех клиниевских номеров всех непустых* *) *функций также рекурсивно перечислимо.*

В самом деле, множество $\boldsymbol{A}$ есть множество решений уравнения $\boldsymbol{K}(y, a) = b$. Ввиду частичной рекурсивности функции $|\boldsymbol{K}(y, a) - b|$ это множество рекурсивно перечислимо (следствие 1 теоремы 1 (п. 6.1)). Аналогично доказывается и рекурсивная перечислимость множества $\boldsymbol{B}$ номеров непустых функций.

Согласно теореме Райса множества $\boldsymbol{A}$ и $\boldsymbol{B}$ нерекурсивны. Таким образом, в дополнение к примеру, построенному в п. 6.3, мы получили серию новых рекурсивно перечислимых нерекурсивных множеств.

7.3. Нумерация Поста. Так как каждое рекурсивно перечислимое множество есть совокупность значений

---

*) Пустой называется функция с пустым графиком, т. е. нигде не определенная функция.

подходящей частично рекурсивной функции и совокупность значений каждой частично рекурсивной функции есть рекурсивно перечислимое множество, то, называя совокупность значений функции $\boldsymbol{K}(n, x)$ при $x = 0, 1, 2, \ldots$ множеством с номером n, мы получим нумерацию всех рекурсивно перечислимых множеств. Эта нумерация называется *нумерацией Поста* и будет далее обозначаться символом $\boldsymbol{\pi}$. В частности, символами $\boldsymbol{\pi}n$ или $\boldsymbol{\pi}_n$ будет обозначаться множество с номером n, т. е. совокупность значений указанной выше функции $\boldsymbol{K}(n, x)$.

Между нумерациями Клини и Поста существует простая связь, часто позволяющая из свойств одной нумерации получать свойства другой. А именно, если $\boldsymbol{A}$ — какое-нибудь рекурсивно перечислимое множество, то постовским номером $\boldsymbol{A}$ является клиниевский номер каждой частично рекурсивной функции, совокупность значений которой совпадает с $\boldsymbol{A}$. Таким образом, если совокупности всех клиниевских номеров отдельных функций называть клиниевскими кирпичами, а совокупности постовских номеров рекурсивно перечислимых множеств — постовскими кирпичами, то каждый постовский кирпич будет блоком, составленным из цельных клиниевских кирпичей. Ясно, что для каждого непустого рекурсивно перечислимого множества $\boldsymbol{A}$ существует бесконечно много частично рекурсивных функций, имеющих $\boldsymbol{A}$ своим множеством значений. Поэтому каждый постовский кирпич, отвечающий непустому множеству, состоит из бесконечного множества клиниевских кирпичей. Исключение составляет лишь постовский кирпич, отвечающий пустому множеству. Он в точности совпадает с клиниевским кирпичом, отвечающим нигде не определенной функции. Отсюда, в частности, следует, что множество постовских номеров всех непустых множеств совпадает с множеством клиниевских номеров всех непустых функций и потому это множество рекурсивно перечислимо.

Рассмотренное соответствие между клиниевскими и постовскими кирпичами показывает также, что теорема Райса справедлива и для постовской нумерации.

Заметим еще, что *совокупность постовских номеров всех множеств, содержащих данное число a, является рекурсивно перечислимой.*

Действительно, совокупность эта совпадает с совокупностью тех значений n, для которых уравнение $\boldsymbol{K}(n, x) = a$ имеет хотя бы одно решение x. Ввиду частичной рекурсивности функции $\boldsymbol{K}$ указанная совокупность рекурсивно перечислима.

Предикат $P(x_1, \ldots, x_n)$ называется рекурсивно перечислимым, если рекурсивно перечислимо множество тех n-ок $\langle x_1, \ldots, x_n \rangle$, для которых предикат P истинен. *Частичной характеристической функцией* предиката P будем называть частичную функцию $\chi(x_1, \ldots, x_n)$, равную 0 на тех n-ках, на которых предикат P истинен, и неопределенную там, где предикат P ложен. Из следствия 3 (п. 6.1) видно, что предикат P тогда и только тогда рекурсивно перечислим, когда его частичная характеристическая функция частично рекурсивна.

Более тонкие и более специфичные свойства рекурсивно перечислимых предикатов дают две следующих теоремы.

Теорема 1 (о представлении предикатов). *Для каждого рекурсивно перечислимого предиката $P(x_1, \ldots, x_{n+1})$ существует такая примитивно рекурсивная функция $h(x_2, \ldots, x_{n+1})$, что*

$$P(x_1, x_2, \ldots, x_{n+1}) \Longleftrightarrow x_1 \in \pi h(x_2, \ldots, x_{n+1}) \qquad (1)$$

для всех значений $x_1, \ldots, x_{n+1}$.

Рассмотрим сначала случай $n = 1$. По условию график предиката $P(x, y)$ можно представить в виде

$$x = v(t), \quad y = w(t) \qquad (t = 0, 1, \ldots), \qquad (2)$$

где v, w — подходящие примитивно рекурсивные функции.

При заданном y все значения t, удовлетворяющие второму из уравнений (2), можно представить (ср. п. 4.2) в виде

$$t = u\,\overline{\mathrm{sg}}\,|w(u) - y| + \mathrm{sg}\,|w(u) - y|\,\mu_z(w(z) = y). \qquad (3)$$

Подставляя вместо t выражение (3) в формулу $x = v(t)$, получим формулу вида $x = F(y, u)$, где F — частично рекурсивная функция от y, u. Функцию $F(y, u)$ можно представить при подходящем числе a в виде $\boldsymbol{K}(a, y, u)$ и, значит,

$$P(x, y) \Longleftrightarrow (\exists u)(\boldsymbol{K}([a, y], u) = x)$$

или

$$P(x, y) \Longleftrightarrow x \in \pi[a, y].$$

Итак, для $n = 1$ теорема 1 доказана. Пусть теперь $n > 1$. Обозначим через $Q(x_1, y)$ предикат $P(x_1, [y]_{n1}, \ldots, [y]_{nn})$. По только что доказанному существует примитивно рекурсивная функция $g(y)$ такая, что

$$Q(x_1, y) \Leftrightarrow x_1 \in \pi g(y).$$

Подставляя сюда вместо y выражение $[x_2, \ldots, x_{n+1}]$, получим требуемое утверждение (1).

Теорему 1 иногда формулируют в следующем виде: *для каждой частично рекурсивной функции $F(x_1, \ldots, x_{n+1})$ существует такая примитивно рекурсивная функция $h(x_2, \ldots, x_{n+1})$, что для любых фиксированных значений $x_2, \ldots, x_{n+1}$ множество значений x, удовлетворяющих уравнению $F(x, x_2, \ldots, x_{n+1}) = 0$, имеет постовский номер $h(x_2, \ldots, x_{n+1})$.*

Чтобы свести эту новую формулировку к старой, достаточно рассмотреть предикат P, определяемый формулой

$$P(x_1, \ldots, x_{n+1}) \Leftrightarrow F(x_1, \ldots, x_{n+1}) = 0,$$

и применить к P теорему 1.

В качестве одного из приложений этой теоремы докажем такое

Следствие. *Существуют такие примитивно рекурсивные функции $u(x, y)$, $v(x, y)$, что для любых чисел m, n*

$$\pi_m \cap \pi_n = \pi_{v(m, n)},$$

$$\pi_m \cup \pi_n = \pi_{u(m, n)}.$$

Согласно определению π_m есть совокупность тех значений t, для которых уравнение $K(m, x) = t$ имеет решение x. Аналогично, π_n есть совокупность тех значений t, для которых уравнение $K(n, y) = t$ имеет какое-то решение y. Отсюда видно, что множество $\pi_m \cap \pi_n$ состоит из тех значений t, для которых уравнение

$$|K(m, x) - t| + |K(n, y) - t| = 0$$

имеет решение $\langle x, y\rangle$. Согласно следствию 4 теоремы о параметризации частично рекурсивных функций (п. 6.1) совокупность P троек $\langle m, n, t\rangle$, для которых указанное уравнение имеет решение $\langle x, y\rangle$, является рекурсивно

перечислимой. По теореме 1 существует примитивно рекурсивная функция $v(m, n)$, для которой

$$\langle m, n, t\rangle \in P \Leftrightarrow t \in \pi_{v(m, n)},$$

но это и значит, что $\pi_m \cap \pi_n = \pi_{v(m, n)}$.

Докажем второе утверждение. Введем функцию

$$G(m, n, x) = \begin{cases} K\left(m, \frac{x}{2}\right), & \text{если } x \text{ четно}, \\ K\left(n, \frac{x-1}{2}\right), & \text{если } x \text{ нечетно}. \end{cases}$$

Согласно теореме о кусочно заданных функциях (п. 6.3) функция $G(m, n, x)$ частично рекурсивна. Множество ее значений при $x = 0, 1, 2, \ldots$ есть $\pi_m \cup \pi_n$. Ищем такое число a, чтобы для всех m, n, x

$$G(m, n, x) = K(a, m, n, x) = K([a, m, n], x).$$

Совокупность значений функции $K([a, m, n], x)$ есть $\pi_{[a, m, n]}$. Таким образом,

$$\pi_m \cup \pi_n = \pi_{[a, m, n]}$$

и, следовательно, в качестве искомой функции $u(m, n)$ можно взять функцию $[a, m, n]$.

В каком-то смысле аналогом для теоремы Клини о неподвижной точке (п. 7.2) является следующая

Теорема 2 (теорема Майхила [57] о неподвижной точке). *Для каждого рекурсивно перечислимого предиката* $P(x_1, \ldots, x_{n+1})$ *существует такая примитивно рекурсивная функция* $g(x_2, \ldots, x_n)$, *что*

$$P(x_1, \ldots, x_n, g(x_2, \ldots, x_n)) \Leftrightarrow x_1 \in \pi g(x_2, \ldots, x_n) \quad (4)$$

для всех значений $x_1, \ldots, x_n$.

В самом деле, согласно теореме 1 найдется такая примитивно рекурсивная функция $h(x_2, \ldots, x_{n+1})$, что будет иметь место соотношение (1). Применяя к функции $h(x_2, \ldots, x_{n+1})$ теорему Клини о неподвижной точке, заключаем, что для подходящей примитивно рекурсивной функции $g(x_2, \ldots, x_n)$ будем иметь

$$\varkappa h(x_2, \ldots, x_n, g(x_2, \ldots, x_n)) = \varkappa g(x_2, \ldots, x_n). \quad (5)$$

Но для любых чисел a, b равенство $\varkappa a = \varkappa b$ влечет равенство $\boldsymbol{\pi} a = \boldsymbol{\pi} b$. Поэтому из (5) имеем

$$\boldsymbol{\pi} h(x_2, \ldots, x_n, g(x_2, \ldots, x_n)) = \boldsymbol{\pi} g(x_2, \ldots, x_n). \quad (6)$$

Наконец, подставляя в соотношение (1) вместо x_{n+1} выражение $g(x_2, \ldots, x_n)$, с помощью равенства (6) получим (4).

В дополнение к теореме о кусочном задании функций из п. 6.3 мы сейчас докажем более общее предложение, полезное при построении частично рекурсивных функций, продолжающих заданные при определенных условиях.

Теорема 3 (о совместном продолжении). *Пусть заданы частично рекурсивные функции* $F_i(x, y)$ *с областями определенности* $\boldsymbol{A}_i$ $(i = 1, \ldots, r)$. *Существует частично рекурсивная функция* $F(x, y, z_1, \ldots, z_r)$, *удовлетворяющая следующим требованиям*:

а) F *определена для тех и только тех значений* x, y, $z_1, \ldots, z_r$, *для которых выполнено хотя бы одно из условий*

$$x \in \boldsymbol{\pi} z_i \,\&\, \langle x, y\rangle \in \boldsymbol{A}_i \qquad (i = 1, \ldots, r);$$

б) *для каждых* x, y, $z_1, \ldots, z_r$ *из области определенности функции* F *существует такое* i, *что* $x \in \boldsymbol{\pi} z_i$ *и*

$$F(x, y, z_1, \ldots, z_r) = F_i(x, y).$$

Функцию F, удовлетворяющую условиям а), б), можно определить, например, следующим алгоритмом. Пусть заданы какие-нибудь числа x, y, $z_1, \ldots, z_r$. По условию существуют регулярные процессы, позволяющие находить элементы множеств $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_r$, $\boldsymbol{\pi}_{z_1}, \ldots, \boldsymbol{\pi}_{z_r}$. Делаем первые шаги в этих процессах, затем последовательно делаем вторые шаги и т. д. В результате некоторых шагов будут получаться элементы упомянутых множеств, а некоторые шаги будут давать промежуточные результаты. Получив после какого-нибудь шага число x из некоторого $\boldsymbol{\pi} z_i$ или пару чисел $\langle x, y\rangle$ из $\boldsymbol{A}_i$, мы проверим, не будет ли среди всех полученных, в том числе и на предыдущих шагах, чисел, пар чисел и информаций вида $a \in \boldsymbol{\pi} z$, $\langle a, b\rangle \in \boldsymbol{A}_j$ для какой-нибудь i-й пары утверждений $x \in \boldsymbol{\pi} z_i$, $\langle x, y\rangle \in \boldsymbol{A}_i$. Как только такая пара найдется, процесс прекращается и вычисляется значение $F_i(x, y)$. Это и будет значением выражения $F(x, y, z_1, \ldots, z_r)$. Если указанный процесс не обрывается, то $F(x, y, z_1, \ldots, z_r)$ считается неопределенным.

Ясно, что построенная таким образом функция F удовлетворяет требованиям теоремы 3. Значения F вычисляются посредством процесса, имеющего алгоритмический характер. В силу тезиса Чёрча (п. 2.3) мы можем верить, что функция F частично рекурсивна. Однако для формального доказательства частичной рекурсивности функции F надо все же выразить F при помощи операций подстановки и μ-операции через рекурсивные функции, то есть переложить приведенное выше определение функции F на язык формул. Это мы теперь и сделаем.

Ищем такие числа $a_1, \ldots, a_r$, чтобы

$$F_i(x, y) = \boldsymbol{K}(a_i, x, y) = \boldsymbol{K}([a_i, x], y) \quad (i = 1, \ldots, r).$$

Представим график функции $w = \boldsymbol{K}(n, y)$ в параметрической форме:

$$n = f(t), \quad y = g(t), \quad w = h(t),$$

где f, g, h — подходящие рекурсивные функции (п. 6.1).

В каждый момент времени t находим числа $f(t)$, $g(t)$, $h(t)$ и запоминаем следующую информацию:

$$h(t) = \boldsymbol{K}(f(t), g(t)), \quad h(t) \in \pi f(t).$$

В соответствии с изложенным ищем наименьшее t, при котором для подходящего i, $1 \leqslant i \leqslant r$, было бы

$$h(u) = x, \ f(u) = z_i, \quad f(v) = [a_i, x], \ g(v) = y,$$

где $u = [t]_{21}$, $v = [t]_{22}$ (см. п. 7.1). Вводя функции

$$\Phi_i(t) = \Phi_i(t, x, y, z_1, \ldots, z_r) =$$
$$= |x - h(u)| + |f(u) - z_i| + |f(v) - [a_i, x]| + |g(v) - y|,$$

мы сможем искомое значение t представить в виде

$$t^* = t^*(x, y, z_1, \ldots, z_r) = \mu_t(\Phi_1 \cdot \ldots \cdot \Phi_r = 0).$$

Зная t^*, мы должны найти еще то значение i, при котором $\Phi_i(t^*) = 0$, после чего положить функцию F равной величине выражения $F_i(x, y) = h(v(t^*))$. Но это будет достигнуто, если мы по определению положим $F(x, y, z_1, \ldots, z_r) = h(v(t^*))$.

Функция F получается из рекурсивных функций f, g, h подстановками, μ-операцией и операциями $+$, $\dot{-}$. Поэтому функция F частично рекурсивна, что нам и требовалось.

В теореме 3 указаны функции $F_i(x, y)$ лишь от двух переменных. Однако ясно, что сама теорема и ее доказательство остаются истинными и в случае, когда задаются функции $F_i(x, y_1, \ldots, y_m)$, содержащие любое число параметров $y_1, \ldots, y_m$.

Заметим еще, что теорема 3 останется верной и в том случае, если в ее формулировке условия $x \in \pi z_i$ заменить условиями $f_i(x, y, z_1, \ldots, z_r) = 0$, где f_i — какие-нибудь частично рекурсивные функции, так как в силу п. 7.3 условия $f_i = 0$ равносильны условиям вида $x \in \pi q_i(y, z_1, \ldots, z_r)$, где q_i — подходящие рекурсивные функции.

7.4. Однозначные нумерации. Нумерации Клини и Поста занимают исключительное место в теории алгоритмов благодаря тому, что, имея какой-нибудь алгоритм для вычисления значений функций $f_0(x)$, $f_1(x)$, ... или чисел множеств S_0, S_1, ..., мы всегда можем найти такую примитивно рекурсивную функцию $\varphi(n)$, что $\varphi(n)$ будет клиниевским номером функции $f_n(x)$ или соответственно постовским номером множества S_n. Конечно, нумерации Клини и Поста — не единственные из нумераций, обладающих этим свойством. Все нумерации, обладающие указанным свойством, носят общее название *геделевских* нумераций. В известном, точно определенном смысле все геделевские нумерации изоморфны друг другу (см. п. 9.3) и все они достаточно сложны. Как мы видели, каждая частично рекурсивная функция и каждое рекурсивно перечислимое множество имеют в нумерациях Клини и Поста нерекурсивные совокупности номеров. Естественно возникает вопрос: нельзя ли построить такую нумерацию всех рекурсивно перечислимых множеств (или частично рекурсивных функций), при которой каждое рекурсивно перечислимое множество (частично рекурсивная функция) имело бы в точности один номер, и в то же время чтобы по номеру n множества (функции) в новой нумерации можно было посредством алгоритма найти постовский (клиниевский) номер $\varphi(n)$ этого множества (функции)? В 1958 г. Фридберг [108] показал, что такие однозначные нумерации рекурсивно перечислимых множеств и всех частично рекурсивных функций действительно существуют. Ниже эти результаты излагаются в слегка обобщенной форме.

Пусть $\mathfrak{S}$ — какая-то непустая система рекурсивно перечислимых множеств. Произвольное отображение α совокупности натуральных чисел N на систему $\mathfrak{S}$ называется нумерацией $\mathfrak{S}$. Через $\alpha(n)$ или αn обозначается то множество из $\mathfrak{S}$, которое отвечает числу n при отображении α. Число n называется *α-номером* множества αn.

Нумерация α называется *вычислимой* (Успенский [103]), если существует общерекурсивная функция $\varphi(x)$ такая, что $\varphi(n)$ есть постовский номер множества αn, то есть если

$$\alpha n = \pi\varphi(n) \qquad (n = 0, 1, \ldots). \tag{1}$$

Система $\mathfrak{S}$ рекурсивно перечислимых множеств называется *вычислимой*, если существует хотя бы одна ее вычислимая нумерация.

Частичная функция $A(n, x)$ называется *универсальной* для нумерации α, если для каждого натурального значения n множество αn совпадает с множеством всех значений $A(n, x)$ при $x = 0, 1, \ldots$

Легко убедиться, что *нумерация α тогда и только тогда вычислима, когда она обладает частично рекурсивной универсальной функцией.*

Действительно, пусть частично рекурсивная функция $A(n, x)$ универсальная для нумерации α. Обозначая через a клиниевский номер функции A, имеем для произвольных n, x

$$A(n, x) = \boldsymbol{K}(a, n, x) = \boldsymbol{K}([a, n], x)$$

и, таким образом, $[a, n]$ — постовский номер множества αn.

Обратно, пусть нумерация α вычислимая и $\varphi(n)$ — общерекурсивная функция, удовлетворяющая соотношению (1). Тогда функция $\boldsymbol{K}(\varphi(n), x)$, очевидно, частично рекурсивна и универсальна для α.

Нумерация α семейства множеств $\mathfrak{S}$ называется *однозначной*, если $\alpha m \neq \alpha n$ для $m \neq n$, то есть если каждое множество из $\mathfrak{S}$ имеет один и только один α-номер.

Поставим вопрос: какие вычислимые семейства множеств обладают вычислимой однозначной нумерацией? Каковы необходимые и достаточные признаки тех вычислимых семейств множеств, которые обладают вычислимыми однозначными нумерациями?

Вопрос этот в настоящее время, по-видимому, открытый. Ниже даются лишь достаточные условия, из которых, однако, следует, что семейство всех рекурсивно перечислимых множеств обладает однозначной вычислимой нумерацией. Предварительно мы докажем следующее очевидное утверждение, справедливое как для вычислимых, так и для однозначных вычислимых нумераций.

Т е о р е м а 1. *Если семейство множеств $\mathfrak{S}$, обладающее (однозначной) вычислимой нумерацией, содержит пустое множество, то, выбросив из $\mathfrak{S}$ это пустое множество, мы получим семейство, допускающее (однозначную) вычислимую нумерацию. Если к семейству множеств $\mathfrak{S}$, обладающему однозначной вычислимой нумерацией, присоединить любое конечное число рекурсивно перечислимых множеств, то получится семейство, обладающее однозначной вычислимой нумерацией.*

В самом деле, пусть $A(n, x)$ — универсальная частично рекурсивная функция, осуществляющая (однозначную) вычислимую нумерацию семейства $\mathfrak{S}$. Обозначим через $\boldsymbol{M}$ совокупность тех значений параметра n, для которых уравнение $A(n, x) = y$ имеет хотя бы одно решение x, y. В силу п. 4.2 множество $\boldsymbol{M}$ рекурсивно перечислимо. Берем общерекурсивную функцию $p(t)$, совокупность значений которой совпадает с $\boldsymbol{M}$ и которая при различных значениях аргумента принимает различные значения (п. 6.3). Ясно, что частично рекурсивная функция

$$A_1(n, x) = A(p(n), x)$$

является универсальной для семейства $\mathfrak{S} - \mathfrak{Q}$. Если нумерация $\mathfrak{S}$ однозначна, то ввиду разнозначности функции p нумерация, осуществляемая функцией $A_1(n, x)$, также однозначная.

Докажем второе утверждение теоремы. Пусть снова $A(n, x)$ — универсальная функция нумерации семейства $\mathfrak{S}$ и пусть $f_0(x), \ldots, f_s(x)$ — примитивно рекурсивные функции, совокупности значений которых $M_0, \ldots, M_s$ различны и не входят в семейство $\mathfrak{S}$. Тогда частично рекурсивная функция

$$A_1(n, x) = \sum_{i=0}^{s} f_i(x)\, \overline{\mathrm{sg}}\, |n - i| + \mathrm{sg}\,(n \dot{-} s)\, A(n \dot{-} (s+1), x),$$

очевидно, является универсальной для семейства $\mathfrak{S}$ и множеств $M_0, \ldots, M_s$. При этом, если функция A давала однозначную нумерацию, то A_1 дает также однозначную нумерацию.

Теорема 2. *Каждая вычислимая нумерация α семейства множеств $\mathfrak{S}$, не содержащего пустого множества, обладает общерекурсивной универсальной функцией.*

Пусть $A(n, x)$ — какая-нибудь частично рекурсивная универсальная для $\mathfrak{S}$ функция. Область определенности этой функции представима в виде совокупности пар $\langle \varphi(t), \psi(t) \rangle$, $t = 0, 1, \ldots$, где φ, ψ — подходящие примитивно рекурсивные функции. Для каждого n совокупность тех значений x, для которых функция $A(n, x)$ определена, можно представить в виде совокупности всех значений, принимаемых при $t = 0, 1, \ldots$ функцией (см. п. 4.2)

$$B(n, t) = \overline{\mathrm{sg}}\, |n - \varphi(t)|\, \psi(t) + \mathrm{sg}\, |n - \varphi(t)|\, \psi(\mu_y(\varphi(y) = n)).$$

Так как для каждого n уравнение $\varphi(y) = n$ имеет решение, то функция $B(n, t)$ общерекурсивна. Функция $A(n, B(n, t))$ универсальна для нумерации α. В силу изложенного она общерекурсивна.

Стандартным номером (γ-номером) конечного множества чисел $m_1, \ldots, m_k$ $(m_1 < m_2 < \ldots < m_k)$ называется число $n = 2^{m_1} + 2^{m_2} + \ldots + 2^{m_k}$. Стандартным номером пустого множества называется 0. Ясно, что стандартная нумерация — одна из вычислимых однозначных нумераций семейства всех конечных множеств. Характерное ее свойство состоит в том, что при ней число элементов множества с номером n, а следовательно, максимальный и минимальный элементы этого множества суть рекурсивные функции от n.

Семейство $\mathfrak{S}$ каких-нибудь конечных множеств называется γ-перечислимым, если совокупность γ-номеров множеств из $\mathfrak{S}$ рекурсивно перечислима. Всякое γ-перечислимое семейство конечных множеств, очевидно, вычислимо и даже однозначно вычислимо.

После этих предварительных замечаний докажем теперь следующую тонкую основную теорему.

Теорема 3. *Пусть вычислимое семейство $\mathfrak{S}$ рекурсивно перечислимых множеств содержит γ-перечислимое подсемейство $\mathfrak{S}_0$ конечных множеств такое, что*

α) *любое конечное подмножество M_0 произвольного множества M из $\mathfrak{S}$ содержится в подходящем конечном подмножестве $M_1 \subseteq M$ и $M_1 \in \mathfrak{S}_0$.*

β) *каждое множество из* $\mathfrak{S}_0$ *содержится в некотором строго большем множестве из* $\mathfrak{S}_0$.

Тогда заведомо существует однозначная вычислимая нумерация семейства $\mathfrak{S}$.

Из теоремы 1 следует, что при доказательстве теоремы 3 мы можем предполагать, что семейство $\mathfrak{S}$ не содержит пустого множества. Согласно теореме 2 в этом случае $\mathfrak{S}$ обладает общерекурсивной универсальной функцией $A(n, x)$ и $\mathfrak{S}$ состоит из множеств $A_0, A_1, \ldots, A_l, \ldots$, где A_l — совокупность значений $A(l, 0)$, $A(l, 1), \ldots$ функции $A(l, x)$.

Наша цель — построить такую последовательность $S_0, S_1, \ldots$ рекурсивно перечислимых множеств, чтобы множества этой последовательности были попарно различны и в совокупности исчерпывали семейство $\mathfrak{S}$. Следуя Фридбергу, мы будем строить эти множества «шагами». Конечная часть множества S_x, построенная после t-го шага, будет обозначаться через S_x^t, а множество S_x будет по определению равно объединению множеств S_x^t $(t = 0, 1, \ldots)$.

Одновременно с множествами S_x^t будут строиться конечные множества A_l^t и особая частичная функция $f(l, t)$.

Множества A_l^t будут строиться так, чтобы выполнялось следующее их свойство:

1) *Каждое непустое* A_l^t *принадлежит* $\mathfrak{S}_0$, $A_l^0 \subseteq A_l^1 \subseteq A_l^2 \subseteq \ldots$ *и* A_l *есть объединение всех* A_l^t.

Если $f(l, t)$ при некоторых значениях l, t определено и $x = f(l, t)$, то мы будем говорить, что число x — последователь числа l в момент t. Если $x = f(l, t-1)$, а $f(l, t)$ не определено, то будем говорить, что x *освобождается* (от l) в момент t. Соотношение $x = f(l, t)$ содержательно означает, что в момент t мы «хотим» далее так строить множества S_x^u, чтобы оказалось $S_x = A_l$, и только некоторые «препятствия» могут нас заставить далее отказаться от этого намерения и в некоторый момент освободить x от l. Числа, не являющиеся в момент t последователями, будем называть *свободными* в момент t. Наконец, будем говорить, что в момент t обозревается номер $l_t = l(t)$, где $l(t)$ — левая нумерационная функция из п. 3.3. Из свойств этой функции следует, что каждый данный номер $n = l_t$ обозревается в бесконечное число моментов t, так как уравнение $n = l_t$ имеет бесконечно много решений для t.

Указываемый ниже процесс построения множеств A_l^t, S_x^t и функции f таков, что означенные множества и функция f будут, помимо требования 1), удовлетворять также требованиям:

2) *В каждый момент* t *произвольное число* x *может быть последователем не более одного числа* l. *Если число* x *в момент* t *освобождается, то в момент* t *и во все следующие моменты* x *остается свободным.*

3) *Для каждого* x $S_x^0 \subseteq S_x^1 \subseteq S_x^2 \subseteq \ldots$ *Если ранее момента* t *число* x *было свободным все время, то* $S_x^{t-1} = \varnothing$.

4) *Для каждых* x, t *из* $S_x^t \neq \varnothing$ *следует* $S_x^t \in \mathfrak{S}_0$. *Если для некоторых* x, t $x = f(l_t, t)$, *то* $S_x^{t-1} \subseteq A_{l_t}^t$.

5) *В каждый момент* t *из* $S_x^t \neq \varnothing$, $x \neq y$, *следует* $S_x^t \neq S_y^t$.

По определению полагаем множества A_l^{-1}, S_x^{-1} пустыми и значения $f(l, -1)$ неопределенными $(l, x = 0, 1, \ldots)$.

Берем произвольное $t \geqslant 0$ и предполагаем, что A_l^u, S_x^u, $f(l, u)$ для всех $u < t$ и всех l, x известны и удовлетворяют требованиям 1) — 5).

Построим сначала множества A_l^t. Полагаем $A_v^t = \varnothing$ для $v > t$. Если же $v \leqslant t$, то конструируем A_v^t следующим образом. По условию существует такая примитивно рекурсивная функция $g(n)$, что числа $g(0), g(1), \ldots, g(z), \ldots$ являются стандартными номерами всех множеств семейства $\mathfrak{S}_0$. Вычисляя постепенно элементы этих множеств и сравнивая для каждого $z = 0, 1, \ldots$ множества $\gamma g(0)$, $\gamma g(1)$, $\ldots$, $\gamma g(z)$ с данными множествами A_v^{t-1}, $\{A(v, 0), A(v, 1), \ldots, A(v, t)\}$ и растущим множеством $\{A(v, 0), A(v, 1), \ldots, A(v, z)\}$, ищем такое значение для s, чтобы

$$\{A(v, 0), \ldots, A(v, t)\} \cup A_v^{t-1} \subseteq \gamma g(s) \subseteq \{A(v, 0), A(v, 1), \ldots, A(v, z)\}.$$

Из предположения α) доказываемой теоремы следует, что искомые значения для s найдутся. Берем минимальное значение s и полагаем $A_v^t = \gamma g(s)$. Ясно, что построенные таким способом множества A_v^t удовлетворяют требованию 1).

Переходим к определению S_x^t и $f(l, t)$. Здесь придется различать несколько случаев.

Случай I: $f(l_t, t-1) = y$ и для некоторого $l < l_t$

$$\{0, 1, \ldots, y\} \cap A_l^t = \{0, 1, \ldots, y\} \cap A_{l_t}^t.$$

Освобождаем y, полагая $f(l_t, t)$ неопределенным, и полагаем $f(l', t) = f(l', t-1)$ $(l' \neq l_t)$, $S_x^t = S_x^{t-1}$.

Случай II: случай I места не имеет и существует число x, для которого $A_{l_t}^t = S_x^{t-1}$ и одновременно

а) $x = f(l, t-1)$ и $l \leqslant l_t$

или

б) x в момент $t-1$ свободно и $x \leqslant l_t$,

или

в) x в момент $t-1$ свободно и ранее момента t

число x *смещено* числом, равным l_t (операция смещения определена ниже (см. случай III В)). В этом случае ничего не меняем, т. е. полагаем $S_x^t = S_x^{t-1}$ и $f(l, t) = f(l, t-1)$ для всех x и l.

Случай III: ни случай I, ни случай II места не имеют. Тогда выполняем последовательно следующие операции:

А) Если $f(l_t, t-1)$ определено, то полагаем $f(l_t, t) = y = f(l_t, t-1)$. Если же $f(l_t, t-1)$ не определено, то через y обозначаем наименьшее из чисел, не бывших до момента t ни разу последователями, и полагаем $f(l_t, t) = y$.

Б) Полагаем $S_y^t = S_y^{t-1} \cup A_{l_t}^t$. В силу свойств 3), 4) или $S_y^{t-1} = \varnothing$, или $S_y^{t-1} \subseteq A_{l_t}^t$. Поэтому после выполнения операции Б) мы будем иметь $S_y^t = A_{l_t}^t$.

В) Если существует такое x (единственное в силу 5)), что $S_x^{t-1} = A_{l_t}^t$ и $x \neq y$, то полагаем $S_x^t = \gamma g(s)$, где s — наименьшее, при котором $\gamma g(s)$ содержит S_x^{t-1} и отлично от S_y^t, S_0^{t-1}, S_1^{t-1}, ... Согласно условию β) такое s заведомо существует.

Выполняя операцию В), мы говорим, что *смещаем* число x числом l_t в момент t.

Г) Если число x, смещаемое числом l_t, в момент $t-1$ не свободно, то освобождаем x.

Полагаем $S_z^t = S_z^{t-1}$, $f(l, t) = f(l, t-1)$ для тех z, l, которые не упоминаются в А), Б), В), Г).

Итак, способ построения S_x^t, $f(l, t)$ указан. Из него непосредственно следует, что множества S_x^t, A_l^t и функция $f(l, t)$ удовлетворяют требованиям 1) — 5). Докажем, что эти множества и функция обладают также следующими свойствами 6)—9).

6) *Каждое число x может смещаться лишь конечное число раз.*

Пусть x смещается в момент t и не свободно в момент $t-1$. Тогда в силу Г) оно будет свободно в момент t, а в силу 2) x будет свободным и во все последующие моменты. Пусть x смещается снова в какой-нибудь из этих последующих моментов $u > t$ числом l_u. По условию x в момент $u-1$ свободно. Если бы было $x \leqslant l_u$ или x смещался числом l_u ранее момента u, то у нас был бы случай IIб) или IIв), что невозможно. Таким образом, во все последующие моменты u число x может смещаться лишь числами l_u, меньшими x, и каждым не более одного раза. Следовательно, x смещается не более конечного числа раз.

7) *Каждое S_x содержится в $\mathfrak{S}$. Если S_x бесконечно, то число x, начиная с некоторого момента t, постоянно является последователем некоторого a и $S_x = A_a$.*

Сначала предположим, что S_x конечно. Тогда для некоторого t $S_x = S_x^t \in \mathfrak{S}_0$ и потому $S_x \in \mathfrak{S}$.

Пусть теперь S_x бесконечно. Так как S_x есть объединение возрастающей последовательности конечных множеств S_x^t ($t = 0, 1, \ldots$), то в этой последовательности бесконечно много раз должен встречаться случай, когда $S_x^{t-1} \neq S_x^t$. Но это возможно лишь либо в случае IIIВ) ($x = y$), либо когда x в момент t смещается числом l_t.

Согласно свойству 6) число x может смещаться лишь конечное число раз. Поэтому для бесконечно многих t должен наступать случай III В), $x = f(l_t, t - 1)$. Но x не может быть последователем различных чисел. Таким образом, для бесконечного числа значений t имеем $a = l_t$, $x = f(a, t - 1)$, $S_x^t = A_a^t$. Следовательно, начиная с некоторого момента t, x — последователь a и $S_x = A_a$.

8) *Если* $x \neq y$, *то* $S_x \neq S_y$.

Пусть, напротив, $x \neq y$, $S_x = S_y$. Если S_x, S_y конечны, то для некоторого t было бы $S_x^t = S_y^t$, что противоречит свойству 5). Пусть S_x, S_y бесконечны. Тогда в силу 7) должны существовать такие числа a, b, что $S_x = A_a$, $S_y = A_b$ и, начиная с некоторого момента t_0, числа x, y — постоянные последователи чисел a, b. Так как $x \neq y$, то $a \neq b$. Пусть $a > b$. Из $A_a = A_b$ следует, что, начиная с некоторого момента t_1,

$$\{0, 1, \ldots, b\} \cap A_a^t = \{0, 1, \ldots, b\} \cap A_b^t \qquad (t \geqslant t_1).$$

Берем такое t, что $l_t = b$, $t > t_0$, $t > t_1$. Тогда в момент t мы имеем случай I и потому в этот момент должны освободить число b, вопреки предположению. Итак, $S_x \neq S_y$ и свойство 8) доказано.

9) *Каждое* A_a *совпадает с некоторым* S_x.

Пусть m — наименьшее число, для которого $A_m = A_a$, и потому для каждого $l < m$ имеем $A_l \neq A_m$. Множества A_l и A_m отличаются некоторым элементом z_l $(l = 0, 1, \ldots, m - 1)$, входящим в одно из них и не входящим в другое. Берем такой момент u, когда каждое из чисел $z_0, z_1, \ldots, z_{m-1}$ уже находится в соответствующем из множеств $A_0^u, A_1^u, \ldots, A_{m-1}^u$. Тогда не только в момент u, но и в произвольные более поздние моменты t_1, t_2 будем иметь

$$A_l^{t_1} \neq A_m^{t_2} \qquad (l < m,\ t_1 \geqslant u,\ t_2 \geqslant u) \tag{2}$$

и, сверх того,

$$\{0, 1, \ldots, x\} \cap A_l^{t_1} \neq \{0, 1, \ldots, x\} \cap A_m^{t_2} \qquad (x \geqslant z), \tag{3}$$

где $z = \max |z_0, z_1, \ldots, z_{m-1}|$.

Допустим, что в какой-то момент t некоторое число c приобретает последователя x, раньше последователем не бывшего. Поскольку последователи приобретаются только посредством операции III А), то

$$l_{t_0} = c, \ x = f(c, t_0), \ S_x^{t_0} = A_c^{t_0}.$$

Посмотрим, как меняется S_x^t при возрастании t от t_0 до момента v, в который x освобождается. Изменение множества S_x^t может происходить лишь при помощи операции IIIБ) и операции смещения x, во время которой происходит одновременно и освобождение x. Поэтому до момента освобождения x множество S_x^t может увеличиваться лишь при помощи операции III Б), то есть в те моменты t, когда

$$l_t = c, \quad x = f(c, t), \quad S_x^t = A_c^t.$$

Таким образом, если t растет от t_0 до v, то множество S_x^t поочередно становится равным $A_c^{t_0}, A_c^{t_1}, \ldots, A_c^{t_k}$ и перед моментом освобождения v

$$S_x^{v-1} = A_c^{t_k} \qquad (t_k \geqslant t_0). \tag{4}$$

Может ли число m бесконечно много раз приобретать и потом терять последователей? Произвольное число, побывав последователем и освободившись, вновь последователем стать не может. Поэтому, если ответ на поставленный вопрос положительный, то число m в какой-то момент $t_0 > u$ должно приобрести последователя $x > z$ и затем в некоторый момент $v > u$ потерять его. Покажем, что это невозможно.

Освобождение x может происходить двумя способами: после операции смещения x числом l_v и при помощи случая I, когда $m = l_v$. Так как

$$x > z, \quad x = f(m, v-1), \quad v \geqslant u,$$

то из неравенств (2) вытекает, что освобождение при помощи случая I невозможно. Допустим, что x смещается в момент v числом l_v и потому

$$S_x^{v-1} = A_{l_v}^v, \quad x = f(m, v-1), \quad l_v < m.$$

Сравнивая эти условия с равенством (4) для $c = m$, получаем

$$A_{l_v}^v = A_m^{t_k} \qquad (t_k \geqslant t_0 > u, \quad v > u, \ l_v < m),$$

что противоречит соотношению (2).

Итак, ответ на заданный вопрос отрицателен и потому *существует такой момент w, начиная с которого m не может терять последователей.*

Существует бесконечно много значений t, удовлетворяющих условиям $l_t = m$, $t \geqslant w$. В каждый из таких моментов t число m не может терять последователя и потому в момент t случай I заведомо не наступает. Допустим, что в рассматриваемые моменты бесконечно часто наступает случай III, то есть для бесконечно многих значений t

$$m = l_t, \ x = f(m, t-1), \ S_x^t = A_m^t \qquad (t \geqslant w).$$

Так как, начиная с момента w, число m не может менять последователей, то число x здесь от t не зависит. Но если для фиксированных x, m равенство $S_x^t = A_m^t$ выполняется при бесконечно многих значениях t, то $S_x = A_m$ и, следовательно, утверждение 9) при сделанных допущениях истинно.

Допустим теперь, что случай III выполняется лишь для конечного числа рассматриваемых значений t. Тогда бесконечно часто должен встречаться хотя бы один из случаев IIа), IIб), IIв).

Пусть бесконечно часто встречается случай IIб), при котором

$$S_x^{t-1} = A_m^t, \qquad x \leqslant m.$$

Число x здесь может меняться с ростом t, но множество возможных значений для него лишь конечно. Поэтому для какого-то фиксированного x равенство $S_x^{t-1} = A_m^t$ будет иметь место при бесконечном числе значений t и потому $S_x = A_m$.

Пусть бесконечно часто наступает случай IIв), когда

$$S_x^{t-1} = A_m^t \text{ и } x \text{ ранее смещен числом } m. \tag{5}$$

Но в момент s смещения x числом m должен наступать случай III, а таких моментов по предположению существует лишь конечное число. Поэтому в соотношении (5) x хотя и зависит от t, но может принимать лишь конечное число значений. Среди этих значений найдется такое x, при котором равенство $S_x^{t-1} = A_m^t$ будет выполняться для бесконечно многих значений t, откуда снова $S_x = A_m$.

Наконец, пусть бесконечно часто наступает случай IIа), при котором

$$S_x^{t-1} = A_m^t, \quad x = f(l, t-1), \quad l \leqslant m, \ t > w. \tag{6}$$

Значение l здесь, вообще говоря, зависит от t, но так как оно не больше фиксированного числа m, то условия (6) должны выполняться бесконечно часто и для некоторого фиксированного значения l, которое далее только и будет рассматриваться.

Пусть $l = m$. Так как после момента w число m не меняет последователей, то x в (6) не зависит от t и потому $S_x^{t-1} = A_m^t$ для бесконечно многих значений t, откуда $S_x = A_m$.

Аналогично, если $l < m$, но x при изменении t принимает лишь конечное множество значений, то тогда снова при некотором фиксированном x для бесконечного числа значений t будем иметь $S_x^{t-1} = A_m^t$, откуда $S_x = A_m$.

Остается рассмотреть случай, когда в соотношении (6) для фиксированного l переменная x принимает бесконечно много различных значений при изменении t. Отсюда следует, что число l бесконечно много раз приобретает последователей.

Пусть x становится последователем l в момент $t_0 > w$ и в момент $t > t_0$ удовлетворяет требованиям (6). Согласно (4) имеем

$$S_x^{t-1} = A_l^{t_k} \qquad (t_k \geqslant t_0).$$

Сравнивая с (6), получаем

$$A_m^t = A_l^{t_k} \qquad (t \geqslant u, \ t_k \geqslant u, \ l < m),$$

что противоречит неравенству (2). Таким образом, рассматриваемый случай невозможен, а во всех предыдущих случаях оказалось, что $A_a = A_m = S_x$ и, значит, свойство 9) истинно.

Свойства 7), 8), 9) в совокупности означают, что отображение $x \to S_x$ является однозначной нумерацией системы $\mathfrak{S}$. Остается доказать вычислимость этой нумерации. Правила построения множеств S_x^t дают возможность для каждых x, t найти стандартный

номер $\varphi(x, t)$ множества S_x^t, причем легко убедиться, что функция $\varphi(x, t)$ общерекурсивная.

С другой стороны, легко строится примитивно рекурсивная функция $\gamma(p, x)$, совокупность значений которой при каждом $p > 0$ совпадает с конечным множеством, имеющим стандартный номер p. Рассмотрим общерекурсивную функцию $\Phi(x, t, y) = \gamma(\varphi(x, t), y)$. Фиксируя x и меняя t, y, получим в качестве значений функции Φ все числа множества S_x и только их. Поэтому функция $\Phi(x, \mathbf{l}(u), \mathbf{r}(u))$ является искомой общерекурсивной функцией, универсальной для нумерации $x \to S_x$, что и требовалось.

Следствие (Фридберг [108]). *Совокупность $\mathfrak{S}$ всех рекурсивно перечислимых множеств обладает однозначной вычислимой нумерацией.*

В самом деле, семейство $\mathfrak{S}$ вычислимо, так как оно имеет универсальную частично рекурсивную функцию $K(n, t)$. С другой стороны, оно содержит все конечные множества, совокупность которых и можно взять в качестве семейства $\mathfrak{S}_0$, указанного в условиях теоремы 3.

Выше рассматривались семейства множеств чисел. Переход от семейств множеств к семействам функций требует лишь нескольких очевидных дополнительных замечаний.

Пусть $\mathfrak{S}$ — семейство одноместных функций. Отображение α: $N \to \mathfrak{S}$ натурального ряда на $\mathfrak{S}$ называется нумерацией $\mathfrak{S}$. Если различным числам отвечают различные функции, то нумерация называется однозначной. Функция $F(n, x)$ называется универсальной для нумерации α, если число n есть α-номер функции $f(x) = F(n, x)$. Нумерации, имеющие частично рекурсивные универсальные функции, называются вычислимыми.

График функции $f(x)$ — это совокупность пар чисел $\langle x, y \rangle$, удовлетворяющих соотношению $f(x) = y$. Совокупность канторовых номеров этих пар условимся временно называть 1-графиком функции f. Согласно п. 6.1 функция f тогда и только тогда частично рекурсивна, когда ее 1-график — множество, рекурсивно перечислимое.

Пусть задана какая-либо нумерация α: $N \to \mathfrak{S}$ некоторого семейства функций $\mathfrak{S}$ и пусть n — α-номер функции $f \in \mathfrak{S}$. Ставя числу n в соответствие 1-график функции f, мы получим нумерацию семейства $\mathfrak{S}_1$ 1-графиков функций из $\mathfrak{S}$. Эту нумерацию семейства $\mathfrak{S}_1$ мы также будем называть α-нумерацией.

Теорема 4. *Семейство функций $\mathfrak{S}$ тогда и только тогда вычислимо (соответственно однозначно вычислимо), когда вычислимо (однозначно вычислимо) семейство $\mathfrak{S}_1$ 1-графиков функций из $\mathfrak{S}$.*

В самом деле, пусть $A(n, t)$ — универсальная функция для семейства $\mathfrak{S}_1$. Фиксируем n. Совокупность A_n значений функции $A(n, t)$ будет 1-графиком некоторой функции $f(x)$ из $\mathfrak{S}$. Найдем эту функцию. Пусть x задано, $y = f(x)$. Тогда номер пары $\langle x, y \rangle$ должен принадлежать A_n, то есть для подходящего t

$$\mathbf{c}(x, y) = A(n, t), \quad x = \mathbf{l}(A(n, t)), \quad y = \mathbf{r}(A(n, t)).$$

Беря для t наименьшее значение, удовлетворяющее второму равенству, получим

$$y = \mathbf{r}(A(n, \mu_t(A(n, t) = x))) = B(n, x).$$

Таким образом, функция $B(n, x)$ является частично рекурсивной универсальной функцией для семейства $\mathfrak{S}$.

Аналогично доказывается и обратное утверждение.

Частичная функция называется *конечно определенной*, если она имеет определенные значения лишь для конечного числа значений аргумента. Ясно, что частичная функция тогда и только тогда конечно определенная, когда ее 1-график является конечным множеством. *Стандартным номером* конечно определенной функции будем называть стандартный номер ее 1-графика. Семейство $\mathfrak{S}_0$ конечно определенных функций называется γ-перечислимым, если совокупность стандартных номеров функций из $\mathfrak{S}_0$ рекурсивно перечислима.

Из теоремы 3 непосредственно вытекает

Теорема 5. *Пусть вычислимое семейство $\mathfrak{S}^*$ частично рекурсивных одноместных функций содержит γ-перечислимое подсемейство $\mathfrak{S}_0^*$ конечно определенных функций такое, что*

α) *любая конечно определенная функция $f_0(x)$, содержащаяся *) в произвольной функции $f(x)$ из $\mathfrak{S}^*$, содержится в подходящей конечно определенной функции $f_1(x)$, содержащейся в функции $f(x)$ и входящей в подсемейство $\mathfrak{S}_0^*$;*

β) *каждая функция из $\mathfrak{S}_0^*$ содержится в некоторой отличной от нее функции из $\mathfrak{S}_0^*$.*

Тогда однозначная вычислимая нумерация семейства $\mathfrak{S}^*$ заведомо существует.

Для доказательства достаточно вместо семейств функций $\mathfrak{S}^*$, $\mathfrak{S}_0^*$ рассмотреть семейства $\mathfrak{S}$, $\mathfrak{S}_0$ их 1-графиков и воспользоваться теоремами 3 и 4.

Как и в случае теоремы 3, из теоремы 5 вытекает главное

Следствие (Фридберг [108]). *Семейство $\mathfrak{S}$ всех одноместных частично рекурсивных функций обладает однозначной вычислимой нумерацией.*

Действительно, семейство $\mathfrak{S}$ вычислимо, так как функция Клини $K(n, x)$ является для него универсальной. С другой стороны, $\mathfrak{S}$ содержит семейство $\mathfrak{S}_0$ всех конечно определенных функций, заведомо обладающее свойствами α), β), указанными в теореме 5. Легко доказывается, что $\mathfrak{S}_0$ γ-перечислимо и потому $\mathfrak{S}$ обладает однозначной вычислимой нумерацией.

Дополнения, примеры и задачи

1. Показать, что значения выражений $[a, x]$, $[x, x]$ являются монотонно возрастающими функциями от x.

2. Используя предыдущий результат, доказать следующее уточнение теоремы Майхила о неподвижной точке: для каждого рекурсивно перечислимого предиката $P(x, y, z)$ существует такая монотонно возрастающая примитивно рекурсивная функция $g(y)$, что $P(x, y, g(y)) \Longleftrightarrow x \in \pi g(y)$.

---

*) Функция $f(x)$ содержится в $g(x)$, если график $f(x)$ содержится в графике функции $g(x)$.

3. Доказать существование примитивно рекурсивной функции $\varphi(m, n, u, v)$ такой, что если функции $f_1(x)$, $f_2(x)$, $g_1(x)$, $g_2(x)$ имеют клиниевские номера m, n, u, v, то $\varphi(m, n, u, v)$ будет клиниевским номером функции $h(x)$, определенной в теореме 4 из п. 6.3 для $n = 2$, $s = 1$.

4. Показать, что объединение конечного числа вычислимых семейств множеств есть вычислимое семейство.

5. Семейство всех рекурсивно перечислимых подмножеств данного рекурсивно перечислимого множества и семейство всех частично рекурсивных функций, содержащихся в заданной частично рекурсивной функции, являются вычислимыми.

6. Семейство всех рекурсивно перечислимых множеств, отличных от заданного рекурсивно перечислимого множества, и семейство всех частично рекурсивных (одноместных) функций, отличных от заданной частично рекурсивной функции, вычислимы.

7. Если семейство рекурсивно перечислимых множеств состоит лишь из бесконечных множеств и содержит все бесконечные рекурсивные множества, то оно не вычислимо (см. Деккер и Майхил [30], Мальцев [52]).

8. Совокупность всех нерекурсивных рекурсивно перечислимых множеств невычислима (Деккер и Майхил [30]).

9. Семейство рекурсивно перечислимых множеств называется вполне перечислимым, если совокупность всех постовских номеров всех множеств этого семейства является рекурсивно перечислимой. Аналогично, семейство частично рекурсивных функций называется *вполне перечислимым*, если совокупность всех клиниевских номеров всех функций этого семейства рекурсивно перечислима.

Показать, что семейство всех надмножеств множеств какой-нибудь γ-перечислимой системы конечных множеств является вполне перечислимым. Аналогично, семейство всех частично рекурсивных расширений функций γ-перечислимой системы конечно определенных функций является вполне перечислимой (Райс [81]).

10. Каждое вполне перечислимое семейство множеств (функций) имеет вид, указанный в предыдущей задаче (см. Райс [81], Деккер и Майхил [30], Мальцев [52]).

§ 8. СВОДИМОСТЬ И КРЕАТИВНОСТЬ МНОЖЕСТВ

С каждым числовым множеством α связана следующая проблема вхождения: найти алгоритм, позволяющий для произвольного числа x узнать, входит x в α или нет. Множества, для которых такой алгоритм существует, были названы в п. 2.3 рекурсивными. Один из обычных методов доказательства рекурсивности некоторого множества α состоит в том, что проблему вхождения для α подходящим образом «сводят» к проблеме вхождения для некоторого другого множества β, рекурсивность которого уже уста-

новлена. Если окажется, что проблема вхождения для множества α «сводится» к проблеме вхождения для множества β и при этом известно, что множество α нерекурсивное, то множество β также будет нерекурсивным. Термин «сводится» можно понимать в разных смыслах. Наиболее простое понятие сводимости было введено Постом [74]. Это понятие и является предметом изучения в данном параграфе. Попутно возникают и изучаются понятия продуктивности и креативности множеств, тесно связанные с нумерацией Поста рекурсивно перечислимых множеств.

8.1. Сводимость и m-эквивалентность множеств. Числовое множество α называется *m-сводимым* к числовому множеству β (символическое обозначение $\alpha \leqslant_m \beta$), если существует такая общерекурсивная функция $f(x)$, что

$$x \in \alpha \Leftrightarrow f(x) \in \beta \qquad (1)$$

для всех значений x. Функция $f(x)$ называется функцией, *m-сводящей* α к β *).

Функция $f(x) = x$ m-сводит каждое множество к самому себе. Далее, если функция f m-сводит множество α к множеству β, а функция g m-сводит β к множеству γ, то функция $g(f(x))$ m-сводит α к γ. Поэтому отношение m-сводимости рефлексивно и транзитивно.

Теорема 1. *Каждое рекурсивное множество α m-сводится к любому непустому множеству β, имеющему непустое дополнение. Если какое-нибудь множество γ m-сводится к рекурсивному или рекурсивно перечислимому множеству δ, то γ рекурсивно или соответственно рекурсивно перечислимо.*

В самом деле, берем какие-нибудь числа $a \in \beta$ и $b \notin \beta$. Функция $f(x)$, равная a на α и b вне α, очевидно, рекурсивна. Она m-сводит α к β и потому первое утверждение теоремы истинно.

Чтобы доказать второе утверждение, обозначим через $\chi(x)$ характеристическую и через $\chi_0(x)$ частичную характеристическую функцию множества δ. Пусть общерекурсивная функция $g(x)$ m-сводит γ к δ. Тогда $\chi(g(x))$ —

*) m-сводимость часто называется также *многосводимостью*. Наряду с *много*сводимостью, в п. 9.2 будет введена еще особая *одно*сводимость.

характеристическая, а $\chi_0(g(x))$ — частичная характеристическая функции множества γ. Если δ рекурсивно, то функция $\chi(x)$, а вместе с нею и функция $\chi(g(x))$ общерекурсивны. Аналогично, если δ рекурсивно перечислимо, то функция $\chi_0(x)$ частично рекурсивная. Но в таком случае и функция $\chi_0(g(x))$ частично рекурсивна, что и требовалось.

Множество β называется *m-универсальным*, если выполнены следующие два условия: 1) β рекурсивно перечислимо; 2) каждое рекурсивно перечислимое множество m-сводится к β.

Иногда говорят, что проблема вхождения для множества α легче проблемы вхождения для множества β, если α m-сводится к β. Тогда можно сказать, что m-универсальные множества — это рекурсивно перечислимые множества, имеющие самую трудную проблему вхождения среди рекурсивно перечислимых множеств. Название «универсальные» происходит от следующего свойства этих множеств. Пусть $f(x)$ — какая-нибудь (частичная) функция, α — числовое множество. Символом $f^{-1}(\alpha)$ обозначают прообраз множества α при отображении $x \to f(x)$ натурального ряда в себя, т. е. $f^{-1}(\alpha)$ — совокупность всех значений x, для которых $f(x) \in \alpha$. Определение m-универсальных множеств теперь можно сформулировать следующим образом: *множество β называется m-универсальным, если оно рекурсивно перечислимо и если каждое рекурсивно перечислимое множество α можно представить в виде $f^{-1}(\beta)$, где $f(x)$ — подходящая общерекурсивная функция.*

Из первоначального определения m-универсальных множеств и транзитивности отношения $\leqslant_m$ непосредственно следует, что *если m-универсальное множество m-сводится к какому-нибудь рекурсивно перечислимому множеству α, то α — также m-универсальное множество.*

Нам известно, что нерекурсивные рекурсивно перечислимые множества существуют. Они m-сводятся к m-универсальным множествам. Значит, в силу теоремы 1 *все m-универсальные множества нерекурсивны.*

Чтобы рассуждения о m-универсальных множествах не были пустыми, надо привести хотя бы один пример m-универсального множества.

Теорема 2. *Совокупность ω канторовских номеров тех пар $\langle a, b\rangle$, для которых разрешимо уравнение $\boldsymbol{K}(a, x) =$*

$= b$, *т. е. для которых* $b \in \pi_a$, *является* $\boldsymbol{m}$*-универсальным множеством.*

Действительно, для произвольного рекурсивно перечислимого множества π_a имеем

$$x \in \pi_a \Longleftrightarrow \boldsymbol{c}(a, x) \in \omega.$$

Поэтому функция $f(x) = \boldsymbol{c}(a, x)$ $\boldsymbol{m}$-сводит π_a к множеству ω. С другой стороны, из частичной рекурсивности функции $\boldsymbol{K}(a, x)$ вытекает, что множество тех пар, для которых уравнение $\boldsymbol{K}(a, x) = b$ имеет решение, рекурсивно перечислимо. Следовательно, рекурсивно перечислимо и множество ω, как совокупность канторовских номеров пар рекурсивно перечислимого множества.

Множества α, β называются *$\boldsymbol{m}$-эквивалентными*, если каждое из них $\boldsymbol{m}$-сводимо к другому.

Из сказанного выше непосредственно вытекают следующие простые свойства $\boldsymbol{m}$-эквивалентности:

а) *Если множество* α *рекурсивно или рекурсивно перечислимо, то рекурсивно или соответственно рекурсивно перечислимо и каждое множество,* $\boldsymbol{m}$*-эквивалентное* α.

б) *Все непустые, отличные от* $\boldsymbol{N}$ *рекурсивные множества* $\boldsymbol{m}$*-эквивалентны друг другу.*

в) *Если множество* α $\boldsymbol{m}$*-универсально, то каждое множество,* $\boldsymbol{m}$*-эквивалентное* α, *также* $\boldsymbol{m}$*-универсально. Все* $\boldsymbol{m}$*-универсальные множества* $\boldsymbol{m}$*-эквивалентны друг другу.*

Так как отношение $\boldsymbol{m}$-эквивалентности транзитивно и симметрично, то все числовые множества распадаются на непересекающиеся классы эквивалентных друг другу множеств. Отношение $\boldsymbol{m}$-сводимости частично упорядочивает эти классы. Если отбросить пустое множество и множество $\boldsymbol{N}$, сами по себе образующие $\boldsymbol{m}$-классы, то среди остальных $\boldsymbol{m}$-классов рекурсивно перечислимых множеств наименьшим в смысле отношения $\leqslant_m$ будет класс рекурсивных, а наибольшим — класс $\boldsymbol{m}$-универсальных множеств.

8.2. Продуктивные и креативные множества. Числовое множество α называется *продуктивным*, если существует такая общерекурсивная функция $f(x)$, что для каждого n

$$\pi_n \subseteq \alpha \Rightarrow f(n) \in \alpha - \pi_n. \tag{1}$$

Функция $f(x)$ называется *продуктивной* для α.

Из этого определения непосредственно видно, что *никакое продуктивное множество не может быть рекурсивно перечислимым*. В частности, каждое продуктивное множество непременно бесконечное.

Действительно, если бы продуктивное множество α совпало с каким-либо рекурсивно перечислимым множеством π_n, то из (1) получилось бы противоречивое утверждение

$$f(n) \in \alpha - \alpha.$$

Один из простейших примеров продуктивных множеств представляет множество δ всех тех чисел x, для которых $x \notin \pi_x$. Действительно, согласно определению

$$n \in \pi_n \Rightarrow n \notin \delta \Rightarrow \pi_n \not\subseteq \delta, \quad n \notin \delta \Rightarrow n \in \pi_n \Rightarrow \pi_n \not\subseteq \delta$$

и потому

$$\pi_n \subseteq \delta \Rightarrow n \notin \pi_n \,\&\, n \in \delta.$$

Таким образом, δ — продуктивное множество с продуктивной функцией $f(x) = x$.

Теорема 1. *Каждое продуктивное множество δ содержит бесконечное рекурсивно перечислимое подмножество.*

Искомое подмножество можно построить следующим путем. Пусть a_0 — какой-либо постовский номер пустого множества и пусть $f(x)$ — продуктивная функция для δ. Так как $\pi_{a_0} \subseteq \delta$, то число $f(a_0)$ содержится в δ. Ищем номер a_1 множества $\{f(a_0)\}$. Из (1) следует, что число $f(a_1)$ содержится в δ и отлично от $f(a_0)$. Ищем номер a_2 множества $\{f(a_0), f(a_1)\}$. Число $f(a_2)$ будет содержаться в δ и будет отлично от $f(a_0)$ и $f(a_1)$. Продолжая этот процесс, получим бесконечную последовательность

$$f(a_0), f(a_1), \ldots, f(a_n), \ldots, \qquad (2)$$

числа которой различны и содержатся в δ. Чтобы это наводящее рассуждение обратить в строгое доказательство, нам надо точнее определить, какой именно номер a_{i+1} множества $\{f(a_0), \ldots, f(a_i)\}$ мы выбираем. Пусть

$$x + 0 \cdot y = \boldsymbol{K}(e, x, y) = \boldsymbol{K}([e, x], y).$$

Поэтому, полагая $[e, f(x)] = g(x)$, имеем

$$\pi_{g(x)} = \{f(x)\}. \qquad (3)$$

В п. 7.3 была построена примитивно рекурсивная функция $u\,(m, n)$, для которой

$$\pi_{u\,(m,\,n)} = \pi_m \cup \pi_n. \qquad (4)$$

Свойства (4) и (3) показывают, что в качестве a_{n+1} можно взять $u\,(a_n, g\,(n))$. Таким образом,

$$a\,(0) = a_0, \quad a\,(n+1) = u\,(a\,(n), g\,(n))$$

и, следовательно, $f\,(a\,(n))$ есть общерекурсивная функция от n. Совокупность (2) всех значений этой функции рекурсивно перечислима, бесконечна и содержится в δ.

Теорема 2. *Если продуктивное множество α $\mathfrak{m}$-сводится к множеству β, то β также продуктивно.*

Обозначим через $f\,(x)$ функцию, сводящую α к β, и через $g\,(x)$ — продуктивную функцию для α. По условию

$$x \in \alpha \Longleftrightarrow f\,(x) \in \beta \Longleftrightarrow x \in f^{-1}\,(\beta), \qquad (5)$$

то есть $\alpha = f^{-1}\,(\beta)$ и, значит,

$$\pi_n \subseteq \beta \Rightarrow f^{-1}\,(\pi_n) \subseteq \alpha. \qquad (6)$$

Мы хотим теперь найти постовский номер множества $f^{-1}\,(\pi_n)$. Это множество есть совокупность тех значений x, для которых $f\,(x) \in \pi_n$. Вводим предикат $P\,(x, y)$:

$$P\,(x, y) \overset{df}{=} f\,(x) \in \pi_y.$$

Согласно теореме о представлении (п. 7.3) существует примитивно рекурсивная функция $h\,(y)$, для которой

$$f\,(x) \in \pi_y \Longleftrightarrow x \in \pi_{h\,(y)}. \qquad (7)$$

Таким образом, множество $f^{-1}(\pi_n)$ есть $\pi_{h\,(n)}$.

Итак, из (6) и продуктивности функции $g\,(x)$ получаем

$$\pi_n \subseteq \beta \Rightarrow \pi_{h\,(n)} \subseteq \alpha \Rightarrow g\,(h\,(n)) \in \alpha - \pi_{h\,(n)}.$$

С другой стороны, из (7) и (5) имеем

$$g\,(h\,(n)) \notin \pi_{h\,(n)} \Longleftrightarrow fgh\,(n) \notin \pi_n,$$
$$g\,(h\,(n)) \in \alpha \Longleftrightarrow fgh\,(n) \in \beta.$$

Следовательно,

$$\pi_n \subseteq \beta \Rightarrow fgh\,(n) \in \beta - \pi_n.$$

Это показывает, что множество β продуктивное с продуктивной функцией $f(g(h(n)))$.

Рекурсивно перечислимое множество α, дополнение которого $\alpha' = \boldsymbol{N} - \alpha$ продуктивно, называется *креативным* *) множеством.

Выше было показано, что совокупность δ тех x, для которых $x \notin \pi_x$, является продуктивной. Ее дополнение δ' состоит из тех x, для которых $x \in \pi_x$, то есть из всевозможных x, для которых разрешимо уравнение

$$\boldsymbol{K}(x, t) = x.$$

Так как левая часть этого уравнения частично рекурсивна, то множество δ' рекурсивно перечислимое и, следовательно, креативное.

Из теоремы 2 непосредственно вытекает такое важное

Следствие 1. *Если креативное множество α $\boldsymbol{m}$-сводится к какому-нибудь рекурсивно перечислимому множеству β, то множество β креативно.*

В самом деле, из $\alpha \leqslant_m \beta$ вытекает $\alpha' \leqslant_m \beta'$. Но α' продуктивно. Поэтому в силу теоремы 2 множество β' также продуктивно, а множество β креативно.

Следствие 2. *Каждое $\boldsymbol{m}$-универсальное множество креативно.*

Действительно, к $\boldsymbol{m}$-универсальному множеству $\boldsymbol{m}$-сводимо любое рекурсивно перечислимое множество, в том числе и креативное множество δ', построенное выше. Ввиду следствия 1 отсюда вытекает креативность $\boldsymbol{m}$-универсального множества.

Более тонко доказывается обращение следствия 2:

Теорема 3 (Майхил [57]). *Каждое креативное множество $\boldsymbol{m}$-универсально.*

Пусть α — заданное креативное множество и $f(x)$ — продуктивная функция для α'. Надо показать, что произвольное рекурсивно перечислимое множество β $\boldsymbol{m}$-сводится к α. Рассмотрим трехместный предикат

$$P(x, y, z) \overset{df}{=} y \in \beta \,\&\, x = f(z).$$

Этот предикат, очевидно, рекурсивно перечислим. Поэтому

*) Иногда его называют *творческим* множеством.

согласно теореме о неподвижной точке для предикатов (п. 7.3) найдется общерекурсивная функция $g(y)$ такая, что

$$y \in \beta \,\&\, x = f(g(y)) \Leftrightarrow x \in \pi_{g(y)}. \tag{8}$$

Пусть $y \in \beta'$. Тогда левая часть соотношения (8) ложна для каждого x и, следовательно, множество $\pi_{g(y)}$ пустое. Если же $y \in \beta$, то отношение $x \in \pi_{g(y)}$ равносильно равенству $x = f(g(y))$ и, следовательно, множество $\pi_{g(y)}$ в этом случае состоит из единственного элемента $f(g(y))$.

Докажем теперь, что функция $f(g(x))$ m-сводит β к α. В самом деле,

$$n \in \beta' \Rightarrow \pi_{g(n)} = \varnothing \Rightarrow \pi_{g(n)} \subseteq \alpha' \Rightarrow f(g(n)) \in \alpha'. \tag{9}$$

С другой стороны, согласно сказанному

$$n \in \beta \Rightarrow \pi_{g(n)} = \{f(g(n))\}. \tag{10}$$

Если бы при этом оказалось, что $f(g(n)) \in \alpha'$, то мы имели бы

$$\pi_{g(n)} \subseteq \alpha' \Rightarrow f(g(n)) \in \alpha' - \pi_{g(n)},$$

что противоречиво. Итак, $f(g(n)) \in \alpha$. Вместе с (9) это приводит к требуемому соотношению

$$n \in \beta \Leftrightarrow f(g(n)) \in \alpha.$$

Теорема 3 доказана. Следствие 2 и теорема 3 показывают, что *класс m-универсальных множеств в точности совпадает с классом креативных множеств.*

В частности, отсюда вытекает, например, что множество δ' тех чисел x, для которых $x \in \pi_x$, не только креативно, что было установлено выше путем непосредственных усмотрений, но и m-универсально, что прямо установить гораздо сложнее.

8.3. Простые множества. Как уже говорилось, при помощи понятия m-сводимости легко доказывается нерекурсивность большого числа множеств. Для этого берется какое-нибудь вспомогательное нерекурсивное множество β и доказывается, что оно m-сводимо к исследуемому множеству α. Тогда α также будет нерекурсивным. Однако этот метод имеет следующий недостаток: если в качестве

вспомогательного множества β мы берем креативное множество, а исследуемое множество α рекурсивно перечислимое, то β будет m-сводиться к α только в случае креативности α. Таким образом, если известный нам запас нерекурсивных множеств состоит лишь из креативных множеств, то при помощи метода m-сведения мы сможем пополнить этот запас только креативными же множествами. В частности, все нерекурсивные рекурсивно перечислимые множества, построенные нами до сих пор, креативны. Естественно возникает вопрос о существовании рекурсивно перечислимых множеств, которые не были бы креативными, но были бы нерекурсивными. Этим требованиям заведомо удовлетворяют так называемые *простые* множества, интересные и с других точек зрения.

Числовое множество α называется *иммунным*, если оно бесконечно и в то же время не содержит никаких бесконечных рекурсивно перечислимых подмножеств. Таким образом, иммунное множество не может быть ни рекурсивно перечислимым, ни продуктивным.

Числовое множество α называется *простым*, если оно само рекурсивно перечислимо, а дополнение его иммунно. Отсюда сразу видно, что простое множество заведомо не рекурсивно и не креативно.

Понятия креативного и простого множества были введены Э. Постом. Ему же принадлежит и следующий пример простого множества.

Рассмотрим общерекурсивную функцию $D(n, x)$, универсальную для одноместных примитивно рекурсивных функций. Эта функция была построена в п. 5.2. Обозначим через δ_n совокупность всех значений функции $D(n, x)$ при $x = 0, 1, 2, \ldots$ Множества δ_n рекурсивно перечислимы и каждое непустое рекурсивно перечислимое множество совпадает с некоторым δ_n. Положим

$$f(x) = \mathbf{r}(\mu_t (D(x, \mathbf{l}(t)) = \mathbf{r}(t) \,\&\, \mathbf{r}(t) > 2x)) =$$
$$= \mathbf{r}(\mu_t (|D(x, \mathbf{l}(t)) - \mathbf{r}(t)| + ((2x+1) \dot{-} \mathbf{r}(t)) = 0)). \quad (1)$$

Таким образом, если значение $f(x)$ определено, то это есть число из δ_x, большее $2x$. При этом, если в δ_x есть числа, бо́льшие $2x$, то $f(x)$ заведомо определено.

Из выражения (1) видно, что функция $f(x)$ частично рекурсивна. Совокупность σ всех значений, принимаемых функцией $f(x)$, есть рекурсивно перечислимое множество. Мы хотим показать, что множество σ простое.

Докажем сначала, что в его дополнении σ' не содержится ни одного бесконечного рекурсивно перечислимого множества. В самом деле, если некоторое δ_n бесконечно, то $f(n)$ определено и $f(n) \in \delta_n$, $f(n) \in \sigma$, откуда $\delta_n \nsubseteq \sigma'$.

Остается доказать, что множество σ' бесконечно. Для этого спросим себя, сколько чисел отрезка $\{0, 1, \ldots, 2n\}$ содержит множество σ? Ясно, что число это не больше числа решений неравенства $f(x) \leqslant 2n$. Но из этого неравенства вытекает $x < n$. Поэтому на отрезке $0, 1, \ldots, 2n$ лежит не более n чисел множества σ и, стало быть, не менее $n+1$ чисел лежит в σ'. Поскольку n произвольное, то σ' бесконечно и, следовательно, σ простое.

8.4. Максимальные множества. Согласно определению дополнение простого множества хотя и бесконечно, но настолько «тесно», что не вмещает ни одного бесконечного рекурсивно перечислимого множества. Усиливая это свойство тесноты (названное в п. 8.3 иммунностью), приходим к следующему понятию сжатых множеств.

Множество U называется сжатым, если

1) *U бесконечно и*

2) *ни для какого рекурсивно перечислимого множества D пересечения $D \cap U$ и $D' \cap U$ не могут быть одновременно бесконечными.*

Условимся говорить, что множества A, B *почти совпадают*, если они отличаются друг от друга лишь конечным числом элементов, то есть если множества

$$A-(A \cap B), \quad B-(A \cap B)$$

конечны. Тогда свойство 2) будет означать, что пересечение U с любым рекурсивно перечислимым множеством либо конечно, либо почти совпадает с U.

Множество M называется максимальным, если оно само рекурсивно перечислимо, а его дополнение сжато.

Из этого определения непосредственно видно, что *рекурсивно перечислимое множество M, имеющее бесконечное дополнение, тогда и только тогда максимально, когда каждое содержащее M рекурсивно перечислимое множество либо почти совпадает с M, либо почти совпадает со всем натуральным рядом.*

Ясно, что каждое максимальное множество простое. Обратное неверно: можно построить простые множества, не являющиеся максимальными (см. задачу 19 в конце этого параграфа). Поэтому вопрос о существовании максимальных множеств имеет существенное значение для теории рекурсивно перечислимых множеств.

Теорема 1 (Фридберг [108]). *Максимальные рекурсивно перечислимые множества существуют.*

Искомое максимальное множество $\boldsymbol{M}$ можно построить следующим путем. Рассмотрим общерекурсивную функцию $D(n, x)$, универсальную для всех одноместных примитивно рекурсивных функций, определенную в п. 5.2. Совокупность всех значений функции $D(n, x)$ (n фиксировано) обозначим через $\boldsymbol{D}_n$. Каждое рекурсивно перечислимое непустое множество совпадает с некоторым $\boldsymbol{D}_n$. Все значения (двуместной) функции $D(n, x)$ расположим в таблицу:

$$\begin{array}{lll} D(0, 0) & & \\ D(0, 1) & D(1, 0) & \\ D(0, 2) & D(1, 1) & D(2, 0) \\ \ldots & \ldots & \ldots \end{array}$$

Множество $\boldsymbol{D}_n$ — это совокупность чисел, лежащих в n-м столбце указанной таблицы. Условимся говорить, что в момент времени $t = 0$ мы вычисляем число $D(0, 0)$, в момент $t = 1$ вычисляем $D(0, 1)$, в момент $t = 2$ вычисляем $D(1, 0)$ и, вообще, в момент t вычисляем $D(\boldsymbol{l}(t), \boldsymbol{r}(t))$, где $\boldsymbol{l}$, $\boldsymbol{r}$ — нумерационные функции из п. 3.3. Мы будем также говорить, что число x в момент t находится в столбце с номером l, если для некоторого $s \leqslant t$

$$x = D(\boldsymbol{l}(s), \boldsymbol{r}(s)), \ \boldsymbol{l}(s) = l.$$

Совокупность всех чисел, находящихся в момент t в столбце с номером l, обозначим через $\boldsymbol{D}_l^t$. Элементы $\boldsymbol{D}_l^t$ — это числа, которые в момент t уже отмечены как входящие в множество $\boldsymbol{D}_l$. Поэтому

$$\boldsymbol{D}_l^0 \subseteq \boldsymbol{D}_l^1 \subseteq \ldots \subseteq \boldsymbol{D}_l^t \subseteq \ldots$$

и $\boldsymbol{D}_l$ есть объединение всех $\boldsymbol{D}_l^t$ ($t = 0, 1, 2, \ldots$). Ясно, что для $l > t$ множества $\boldsymbol{D}_l^t$ пустые.

Произвольное число x в момент t может находиться в одном столбце, или сразу в нескольких столбцах таблицы, или может не находиться ни в одном столбце. Положение x в момент t в таблице мы будем характеризовать числами

$$H_l^t(x) = a_0 \cdot 10^l + a_1 \cdot 10^{l-1} + \ldots + a_l \qquad (l = 0, 1, \ldots),$$

где $a_i = 1$, если $x \in \boldsymbol{D}_i^t$, и $a_i = 0$, если $x \notin \boldsymbol{D}_i^t$.

Число $H_l^t(x)$ мы будем называть *l-высотой* числа x в момент t. Единицы и нули в десятичной записи l-высоты числа x в момент t указывают, входит число x в момент t в соответствующий столбец таблицы или нет. Равная нулю l-высота числа x в момент t означает, что в этот момент x не содержится ни в одном из столбцов с номерами $0, 1, \ldots, l$. Если $x \in \boldsymbol{D}_l^t$, то десятичная запись числа

$H_l^t(x)$ оканчивается цифрой 1 и потому l-высота числа x в этот момент нечетная. Если $x \notin D_l^t$, то l-высота x в момент t четная.

Отметим сразу же следующие очевидные свойства l-высот.

а) *Для любого u*

$$H_l^t(x) \leqslant H_l^{t+u}(x).$$

б) *Для любых l, u*

$$H_l^t(x) < H_l^t(y) \Rightarrow H_{l+u}^t(x) < H_{l+u}^t(y)$$

и, в частности,

$$H_{l+u}^t(x) = H_{l+u}^t(y) \Rightarrow H_l^t(x) = H_l^t(y).$$

Укажем теперь *алгоритм* для построения неубывающей цепочки конечных множеств

$$\boldsymbol{M}^{-1} \subseteq \boldsymbol{M}^0 \subseteq \boldsymbol{M}^1 \subseteq \ldots \subseteq \boldsymbol{M}^t \subseteq \ldots,$$

объединение которых $\boldsymbol{M}$ и окажется максимальным множеством. По определению в качестве $\boldsymbol{M}^{-1}$ берем пустое множество. Допустим теперь, что $\boldsymbol{M}^{t-1}$ построено. Вычисляем число $x = D(\boldsymbol{l}(t), \boldsymbol{r}(t))$ таблицы на стр. 176. Пусть $l = \boldsymbol{l}(t)$. Если

$$x \in \boldsymbol{D}_l^{t-1} \text{ или } x \in \boldsymbol{M}^{t-1},$$

то полагаем $\boldsymbol{M}^t = \boldsymbol{M}^{t-1}$. Если же

$$x \notin \boldsymbol{D}_l^{t-1} \text{ и } x \notin \boldsymbol{M}^{t-1}, \tag{1}$$

то просматриваем все такие числа $f \geqslant l$, для которых $x \in \boldsymbol{D}_f^t$. Поскольку в момент t в таблице столбцы номеров, бо́льших t, пусты, то чисел f, обладающих упомянутыми свойствами, конечное множество. Для каждого из указанных чисел f ищем все такие числа y, для которых существуют числа $y_1, \ldots, y_f$, удовлетворяющие требованиям

$$y_1 < y_2 < \ldots < y_f < y < x, \tag{2}$$

$$y_1 \notin \boldsymbol{M}^{t-1}, \ \ldots, \ y_f \notin \boldsymbol{M}^{t-1}, \ y \notin \boldsymbol{M}^{t-1}, \tag{3}$$

$$H_f^t(y_1) = \ldots = H_f^t(y_f) = H_f^t(y) = H_f^t(x) - 1. \tag{4}$$

Добавляя все числа y для каждого f в множество $\boldsymbol{M}^{t-1}$, получим множество $\boldsymbol{M}^t$. Объединяя все множества $\boldsymbol{M}^t$ $(t = 0, 1, \ldots)$, получим множество $\boldsymbol{M}$. Так как множества $\boldsymbol{M}^t$ строятся посредством алгоритма, то в силу тезиса Чёрча множество $\boldsymbol{M}$ рекурсивно перечислимое. Конечно, это не строгое доказательство. Однако строгое доказательство рекурсивной перечислимости $\boldsymbol{M}$ построить нетрудно и оно предоставляется читателю. Несколько более сложно доказать, что дополнение $\boldsymbol{M}$ обладает свойствами 1), 2).

Вместе с Фридбергом будем говорить, что *число x в момент t обитает на с-высоте i, если*

$$H_c^t(x) = i, \quad x \notin \boldsymbol{M}^{t-1}. \tag{5}$$

Посмотрим, по каким причинам число x, обитающее в некоторый момент t на c-высоте i, в момент $t+1$ перестает обитать на указанной c-высоте. Согласно (5) это может произойти, во-первых, если x в момент $t+1$ имеет бóльшую c-высоту, чем в момент t, то есть в момент $t+1$ вычисляется число $x = D(\boldsymbol{l}(t+1), \boldsymbol{r}(t+1))$ и притом

$$x \notin \boldsymbol{D}_l^t, \qquad l \leqslant c \qquad (l = \boldsymbol{l}(t+1)). \tag{6}$$

Во-вторых, это может произойти потому, что x в момент $t+1$ попадает в $\boldsymbol{M}^{t+1}$ и, значит, в любой следующий момент x не обитает ни на какой высоте.

Число x называется *резидентом* c-высоты i, если, начиная с некоторого момента времени t_0, x постоянно обитает на c-высоте i. Так как в этом случае x не входит ни в какое $\boldsymbol{M}^t$, то $x \in \boldsymbol{M}'$ и, значит, *все резиденты содержатся в* $\boldsymbol{M}'$.

С другой стороны, пусть $x \in \boldsymbol{M}'$ и c — произвольное фиксированное число. Число различных c-высот конечно. С течением времени c-высота x не убывает. Следовательно, начиная с некоторого момента времени, число x будет иметь какую-то постоянную c-высоту i. Так как при любом t $x \notin \boldsymbol{M}^t$, то x — резидент c-высоты i. c-высота i называется *вполне обитаемой*, если в различные моменты времени обитателями этой c-высоты побывает бесконечно много различных чисел, иначе говоря, если для каждого натурального n найдутся такие x, t, что

$$x > n, \qquad x \notin \boldsymbol{M}^{t-1}, \qquad H_c^t(x) = i. \tag{7}$$

В любой момент времени t множество $\boldsymbol{M}^t$ может содержать лишь числа, не превосходящие наибольшего числа, находящегося в данный момент в таблице. Поэтому все числа, бóльшие указанного наибольшего табличного числа, обитают в момент t на c-высоте 0. Таким образом, *c-высота 0 вполне обитаемая.*

Лемма 1. *Для каждого c пусть i_c обозначает наибольшую вполне обитаемую c-высоту. Для каждого c существует не менее c резидентов c-высоты i_c и потому множество $\boldsymbol{M}'$ бесконечно.*

По условию существует лишь конечное множество чисел, которые в тот или иной момент времени будут обитать на c-высоте, большей i_c. Поэтому найдется такой момент времени t_0, начиная с которого каждое число, обитающее в некоторый момент $t_1 > t_0$ на c-высоте, большей или равной i_c, обитает на той же высоте в любой момент $t > t_0$.

c-высота i_c вполне обитаема. С другой стороны, в каждый момент прекращают обитать на указанной c-высоте не более конечного множества чисел. Поэтому существует бесконечно много чисел, которые будут обитать на c-высоте i_c в тот или иной момент времени, больший t_0. Покажем, что *наименьшие c из этих чисел заведомо будут резидентами c-высоты i_c.*

Пусть y — одно из указанных c чисел. Допустим, что y не резидент. По условию, y обитает на c-высоте i_c в какой-то момент $t_1 > t_0$. Перейти в некоторый позднейший момент в обитатели более высокой c-высоты число y не может в силу выбора t_0. Оставаться вечно обита-

телем c-высоты i_c число y не может, так как оно не резидент этой c-высоты. Поэтому найдется момент $t+1>t_1$, начиная с которого y уже не будет обитать на высоте i_c, и потому будем иметь

$$y \notin M^{t-1}, \qquad y \in M^t, \tag{8}$$

$$H_c^t(y) = i_c. \tag{9}$$

Положим $D(l(t), r(t)) = x$, $l(t) = l$. Согласно сказанному выше, из условий (8) следует, что существуют числа $f, y_1, \ldots, y_f$, связанные условиями

$$x \in D_f^t, \quad l \leqslant f,$$

и соотношениями (1), (2), (3), (4).

Пусть $c \leqslant f$. Тогда из (4), б) и (9) следует

$$H_c^t(y_1) = \ldots = H_c^t(y_c) = H_c^t(y) = i_c,$$

то есть в момент t на c-высоте i_c обитает c чисел $y_1, \ldots, y_c$, меньших y. Это противоречит условию, что y — одно из c *наименьших* чисел, побывавших обитателями c-высоты i_c в подходящие моменты времени после t_0.

Пусть, напротив, $f < c$. Из (9), б) и (4) теперь следует

$$H_c^t(y) < H_c^t(x). \tag{10}$$

Так как $x \notin M^{t-1}$, то из (10) видно, что x обитает в момент t на c-высоте, большей i_c. Но тогда число x должно обитать на той же c-высоте и в момент $t-1$, то есть

$$H_c^{t-1}(x) = H_c^t(x). \tag{11}$$

Однако (см. (1)) из

$$x \notin D_l^{t-1}, \quad x \in D_l^t, \quad l \leqslant f < c,$$

следует, что

$$H_c^{t-1}(x) < H_c^t(x),$$

вопреки соотношению (11). Полученное противоречие завершает доказательство леммы 1.

Лемма 2. *Для любого c не более чем одна c-высота имеет бесконечно много резидентов.*

Как и в лемме 1, обозначим через i_c наибольшую вполне обитаемую c-высоту. Всякая c-высота, имеющая бесконечно много резидентов, вполне обитаемая. Поэтому никакая c-высота, бо́льшая i_c, не может иметь бесконечно много резидентов и лемма 2 будет доказана, если удастся показать, что *каждая c-высота i, меньшая i_c, содержит не более c резидентов.*

Допустим, напротив, что c-высота i имеет $c+1$ резидентов $y_1 < \ldots < y_c < y$. Обозначим через t_0 момент времени, начиная с которого все указанные резиденты постоянно обитают на c-высоте i.

Среди чисел, появляющихся в таблице после момента t_0 и в какой-то дальнейший момент обитающих на высоте i_c, ищем число x, большее y.

По условию, в какой-то момент $u > t_0$

$$H_c^u(y) < H_c^u(x), \quad x \notin \boldsymbol{M}^{u-1}. \tag{12}$$

Ищем наименьшее число f, для которого

$$H_f^u(y) \neq H_f^u(x).$$

Из (12) следует, что $f \leqslant c$ и

$$H_f^u(y) < H_f^u(x), \quad x \in \boldsymbol{D}_f^u.$$

Так как для $d < f$ имеем $H_d^u(y) = H_d^u(x)$, то

$$H_f^u(y) = H_f^u(x) - 1.$$

Рассмотрим наименьшее t, для которого $H_f^t(x) = H_f^u(x)$. Согласно (6) в момент t имеем

$$x = \boldsymbol{D}(\boldsymbol{l}(t), \boldsymbol{r}(t)), \quad x \notin \boldsymbol{D}_l^{t-1}, \quad l \leqslant f \quad (l = \boldsymbol{l}(t)).$$

Поскольку x появляется в таблице позже момента t_0, то $t > t_0$ и, значит,

$$H_f^t(y_1) = \ldots = H_f^t(y_c) = H_f^t(x) - 1.$$

Итак, в момент t вычисляется число x, для которого выполнены условия (1) и существуют числа $f, y_1, \ldots, y_f, y$, удовлетворяющие требованиям (2), (3), (4). Поэтому $y \in \boldsymbol{M}^t$ и y не может быть резидентом никакой высоты, вопреки предположению. Лемма 2 доказана.

Нам надо было доказать, что множество $\boldsymbol{M}'$ сжато. Бесконечность $\boldsymbol{M}'$ уже установлена в лемме 1. Поэтому остается доказать, что ни для какого c множества

$$\boldsymbol{M}' \cap \boldsymbol{D}_c, \quad \boldsymbol{M}' \cap \boldsymbol{D}_c'$$

не могут быть одновременно бесконечными. Но уже указывалось, что в любой момент времени элементы из $\boldsymbol{D}_c'$ имеют четную c-высоту, а элементы из $\boldsymbol{D}_c$ имеют нечетную c-высоту. С другой стороны, все элементы из $\boldsymbol{M}$ суть резиденты подходящих c-высот. Поэтому числа из $\boldsymbol{M}' \cap \boldsymbol{D}_c'$ являются резидентами четных высот, а числа из $\boldsymbol{M}' \cap \boldsymbol{D}_c$ — резидентами нечетных высот. Множество c-высот конечно. Если бы оба множества $M' \cap \boldsymbol{D}_c$ и $M' \cap \boldsymbol{D}_c'$ были бесконечными, то существовали бы четная и нечетная c-высоты, каждая из которых имеет бесконечно много резидентов, что противоречит лемме 2.

Итак, $\boldsymbol{M}'$ — сжатое, $\boldsymbol{M}$ — максимальное и теорема 1 доказана.

Дополнения, примеры и задачи

1. Если α — числовое множество, то через $f(\alpha)$ будем обозначать множество значений функции $f(x)$ при $x \in \alpha$. Если общерекурсивная функция $f(x)$ осуществляет одно-однозначное отображение натурального ряда на свою часть, то из креативности (продуктивности) α вытекает креативность (продуктивность) $f(\alpha)$.

2. Если из продуктивного множества вычесть какое-нибудь содержащееся в нем рекурсивно перечислимое множество, то останется снова продуктивное множество. Если к креативному множеству добавить не пересекающееся с ним рекурсивно перечислимое множество, то получится креативное множество.

3. Совокупность постовских номеров одноэлементного множества продуктивна. Совокупность постовских номеров всех рекурсивно перечислимых множеств, отличных от множества $\{0\}$, также продуктивна.

4. Построить продуктивное множество, дополнение которого также продуктивно (см. предыдущую задачу).

5. Каждое бесконечное рекурсивно перечислимое множество содержит продуктивное подмножество и креативное подмножество.

6. Если иммунное множество α m-сводится к множеству β, то β иммунное. Аналогичное верно и для простых множеств.

7. Каждое бесконечное рекурсивно перечислимое множество содержит иммунное и простое подмножества.

8. Рекурсивно перечислимое множество называется *мезоичным*, если оно ни рекурсивное, ни креативное, ни простое. Доказать существование мезоичных множеств (Деккер [28]).

9. Каждое нерекурсивное рекурсивно перечислимое множество есть сумма двух непересекающихся нерекурсивных рекурсивно перечислимых множеств (Фридберг [108]).

10. *Частичной продуктивной функцией* для множества α называется частично рекурсивная функция $p(n)$ такая, что

$$\pi_n \subseteq \alpha \Rightarrow p(n) \text{ определено и } p(n) \in \alpha - \pi_n.$$

Если α имеет частичную продуктивную функцию $p(n)$, то α имеет и общерекурсивную продуктивную функцию, то есть α продуктивно. Действительно, пусть δ — область определенности $p(n)$. Предикат $y \in \delta \,\&\, (x = p(y) \vee x \in \pi_y)$ рекурсивно перечислим. По теореме о представлении (п. 7.3) найдется общерекурсивная функция $f(y)$ такая, что

$$y \in \delta \,\&\, (x = p(y) \vee x \in \pi_y) \Leftrightarrow x \in \pi_{f(y)}.$$

Отсюда следует: 1) $n \in \delta \Rightarrow \pi_{f(n)} = \pi_n \cup \{p(n)\}$; 2) $n \notin \delta \Rightarrow \Rightarrow \pi_{f(n)} = \varnothing$; 3) $n \notin \delta \Rightarrow p(f(n))$ определена. Таким образом, области определенности $p(x)$ и $p(f(x))$ в сумме дают N. Берем общерекурсивную функцию $g(x)$, значения которой частью совпадают с $p(x)$, частью с $p(f(x))$ (см. п. 7.3, теорема 3). Так как p и pf частично продуктивные для α, то g будет искомой продуктивной функцией для α.

11. Подмножество β продуктивного множества α называется *продуктивным центром* α, если $\beta = \{p(x) \mid \pi_x \subseteq \alpha\}$ для подходящей частичной продуктивной функции $p(x)$ для α. Показать, что если β — продуктивный центр продуктивного множества α, то α обладает и таким продуктивным центром γ, что $\gamma \subset \delta$ и $\delta - \gamma$ бесконечно. Показать, что каждое продуктивное множество имеет любое конечное число попарно не пересекающихся продуктивных центров.

12. Множество α тогда и только тогда продуктивно, когда существует такая общерекурсивная функция $f(x)$, что

$$f(x) \in (\alpha - \pi_x) \bigcup (\pi_x - \alpha)$$

для всех значений x (Майхил, ср. Фридберг и Роджерс [109]).

13. Пересечение продуктивного и простого множеств есть множество продуктивное (Деккер [29]).

14. Анализ доказательства теоремы 3 из п. 8.2 непосредственно приводит к следующим результатам, несколько более тонким, чем сама теорема 3. Множество α назовем *слабо продуктивным*, если для него существует такая общерекурсивная функция $f(x)$, что

а) $\pi_x = \varnothing \Rightarrow f(x) \in \alpha$,

б) (π_x одноэлементно и $\pi_x \subseteq \alpha) \Rightarrow f(x) \in \pi_x$.

Показать, что каждое рекурсивно перечислимое множество m-сводимо к дополнению произвольного слабо продуктивного множества и потому каждое слабо продуктивное множество продуктивно.

15. Пусть все множества некоторого семейства $\mathfrak{S}$ имеют одну и ту же слабо продуктивную функцию. Тогда

а) существует функция, продуктивная одновременно для каждого множества из $\mathfrak{S}$;

б) пересечение всех множеств из $\mathfrak{S}$ продуктивно;

в) объединение всех множеств из $\mathfrak{S}$ продуктивно (Шмульян [121], стр. 101).

16. Каждое рекурсивно перечислимое множество, имеющее бесконечное дополнение, содержится в некотором простом множестве. Верно ли, что каждое рекурсивно перечислимое множество, имеющее бесконечное дополнение, содержится в некотором максимальном множестве?

17. Согласно п. 7.4 γ-номером (стандартным номером) конечного множества, состоящего из чисел $c_1 < c_2 < \ldots < c_m$, называется число $2^{c_1} + 2^{c_2} + \ldots + 2^{c_m}$, а γ-номером пустого множества называется 0. Словами γn, πn обозначаются конечное множество, имеющее стандартный номер n, и соответственно рекурсивно перечислимое множество, имеющее постовский номер n.

Бесконечное множество α называется *гипериммунным*, если не существует ни одной общерекурсивной функции $i(x)$, для которой все множества $\gamma i(0), \gamma i(1), \ldots, \gamma i(n), \ldots$ попарно не пересекаются и каждое имеет хотя бы один элемент из α.

Бесконечное множество α называется *гипергипериммунным*, если не существует ни одной общерекурсивной функции $i(x)$, для

которой множества $\pi i(0)$, $\pi i(1)$, ..., $\pi i(n)$, ... попарно не пересекаются, конечны и каждое из них имеет непустое пересечение с α.

Множество α называется *гиперпростым* или *гипергиперпростым*, если α рекурсивно перечислимое, а его дополнение α' гипериммунное или соответственно гипергипериммунное.

Показать, что каждое гипергиперпростое множество гиперпросто, а гиперпростое просто. Доказать, что все максимальные множества гипергиперпростые (Фридберг [108]).

18. Построить максимальные множества α, β, отличающиеся бесконечным множеством элементов (например, принять во внимание, что дополнение максимального множества состоит либо почти целиком из четных, либо почти целиком из нечетных чисел).

19. Если максимальные множества α, β отличаются бесконечным множеством чисел, то их пересечение гипергиперпростое, но не максимальное (Ейтс [36]).

20. Если рекурсивно перечислимое множество α не гипергиперпростое и его дополнение α' бесконечное, то найдется такая общерекурсивная функция $j(x)$, что последовательность $\pi j(0)$, $\pi j(1)$, ... будет состоять из попарно не пересекающихся множеств, каждое из которых имеет бесконечное пересечение с α' (Ейтс [36]).

21. Существует гиперпростое множество, не являющееся гипергиперпростым.

22. *Прямым пересчетом* (или трассирующей функцией) множества α называется функция $f(x)$, пересчитывающая элементы α строго в порядке их возрастания. Иначе говоря, для бесконечного α $f(x)$ — монотонно возрастающая функция, совокупность значений которой совпадает с α. Показать, что общерекурсивным прямым пересчетом обладают бесконечные рекурсивные множества и только они.

23. Говорят, что функция $g(x)$ *мажорирует* функцию $f(x)$, если существует такое a, что $g(x) \geqslant f(x)$ для $x \geqslant a$. Доказать, что рекурсивно перечислимое множество α тогда и только тогда гиперпростое, когда его дополнение α' бесконечно и прямой пересчет α' не мажорируется никакой общерекурсивной функцией (Успенский [104], Райс [83]).

24. Говорят, что частично рекурсивная функция $f(x)$ *ретрассирует* множество α, если для каждого $x \in \alpha$: а) значение $f(x)$ определено; б) если число x не минимальное в α, то $f(x)$ есть ближайшее меньшее в α; в) если x минимальное в α, то $f(x) = x$. Множество α называется *ретрассируемым*, если существует хотя бы одна частично рекурсивная функция, ретрассирующая α.

Пусть числа a_0, a_1, a_2, ..., a_n, ... удовлетворяют условиям: $1 \leqslant a_0 \leqslant 9$, $0 \leqslant a_i \leqslant 9$ $(i = 1, 2, \ldots)$. Показать, что множество

$$\sigma = \{a_0,\ 10\cdot a_0 + a_1,\ 10^2\cdot a_0 + 10\cdot a_1 + a_2,\ \ldots\}$$

ретрассируется функцией

$$f(x) = \begin{cases} x, & \text{если } 0 \leqslant x \leqslant 9, \\ \left[\dfrac{x}{10}\right], & \text{если } 10 \leqslant x. \end{cases}$$

Таким образом, функция $f(x)$ является ретрассирующей для континуального числа множеств σ.

Показать, что каждое рекурсивное множество ретрассируемо и что каждое ретрассируемое множество рекурсивно или иммунно (Деккер и Майхил [31]).

25. Каждое ретрассируемое множество, имеющее рекурсивно перечислимое дополнение, или рекурсивно, или гипериммунно (см. задачу 23) (Деккер и Майхил [31]).

26. Рекурсивно перечислимое множество α тогда и только тогда имеет ретрассируемое дополнение, когда существует общерекурсивная функция $\chi(x)$ такая, что множество α является совокупностью тех значений x, для которых неравенства

$$\chi(y) \leqslant \chi(x) \ \& \ y > x$$

имеют некоторое решение y (*представление* Ейтса [36]).

27. Множество α тогда и только тогда гипергиперпростое, когда оно рекурсивно перечислимо, его дополнение α бесконечно и α' не содержит бесконечных ретрассируемых подмножеств (Ейтс [36]).

28. *Табличная сводимость.* С алгоритмической точки зрения m-сводимость множества α к множеству β означает истинность для некоторой общерекурсивной функции $f(x)$ следующего предписания: если хотите знать, расположено число x внутри или вне множества α, то найдите $f(x)$ и узнайте расположение $f(x)$ относительно β. Если $f(x)$ внутри β, то x внутри α; если же $f(x)$ вне β, то x вне α.

Вид этого предписания легко обобщается, например, следующим образом: если хотите знать, как расположено x относительно α, вычислите значения двух рекурсивных функций $f_1(x)$, $f_2(x)$ и посмотрите, как эти значения расположены относительно β. Если $f_1(x) \in \beta \ \& \ f_2(x) \in \beta$, то $x \in \alpha$, а в остальных случаях $x \notin \alpha$.

Последний критерий можно изменить, потребовав, чтобы $x \in \alpha$ было в случаях $f_1(x) \notin \beta \ \& \ f_2(x) \in \beta$ или $f_1(x) \in \beta \ \& \ f_2(x) \notin \beta$, а в остальных случаях чтобы было $x \notin \alpha$.

Класс рассматриваемых предписаний можно еще расширить путем задания n функций $f_1(x), \ldots, f_n(x)$ и усложнения соответствующих критериев вхождения x в α (таблиц истинности). Все эти предписания были названы Постом [74] *табличными сводимостями*. Ясно, что вместо задания n функций можно ограничиться и заданием лишь одной функции $x(t) = [n-1, [f_1(t), \ldots, f_n(t)], m-1]$. Тогда знание $x(t)$ дает не только число n, значения $f_1(t), \ldots, f_n(t)$, но и номер m соответствующей таблицы истинности. Изложенное можно резюмировать в следующем формальном определении.

Условимся говорить, что число x *таблично сводимо* к множеству β (символически $xtt\beta$), если в результате следующих вычислений:

1) ищем числа $n = [x]_{31} + 1$, $y = [x]_{32}$, $m = [x]_{33} + 1$, где $[x]_{ij}$ — нумерационные функции из п. 7.1 (или соответствующие из п. 3.3);

2) ищем числа $\varepsilon_1 = \chi([y]_{n1}), \ldots, \varepsilon_n = \chi([y]_{nn})$, где χ — характеристическая функция множества β;

3) находим число $u = \varepsilon_1 + \varepsilon_2 \cdot 2 + \ldots + \varepsilon_n \cdot 2^{n-1}$ и, представляя m в виде $m = 2^{u_0} + 2^{u_1} + \ldots + 2^{u_s}$ $(u_0 < u_1 < \ldots \ldots < u_s)$, находим числа $u_0, u_1, \ldots, u_s$

— окажется, что $u \in \{u_0, u_1, \ldots, u_s\}$.

Множество α называется таблично сводимым к множеству β (символически $\alpha tt\beta$), если существует такая общерекурсивная функция $f(x)$, что $x \in \alpha \iff f(x)\, tt\beta$ для каждого значения x.

Показать, что отношение табличной сводимости рефлексивно, транзитивно и что из m-сводимости вытекает табличная сводимость.

29. Говорят, что множество α сводится к β таблицами ширины w, если существует такая общерекурсивная функция $f(x)$, что $x \in \alpha \iff f(x)\, tt\beta$ и $[f(x)]_{31} + 1 \leqslant w$ для каждого значения x.

Если множество α сводится к β таблицами ширины v, а β сводится к γ таблицами ширины w, то α сводится к γ таблицами ширины vw. Из m-сводимости вытекает сводимость таблицами ширины 1. Верно ли обратное?

30. Если α таблично сводимо к рекурсивному множеству, то α рекурсивное.

31. Креативные множества таблично сводимы к простому множеству Поста, построенному в п. 8.3 (П о с т [74]).

32. Креативные множества не сводимы таблично к гиперпростым множествам (П о с т [74]).

33. Множество α называется *ограниченно таблично сводимым* к множеству β, если для некоторого w α сводимо к β таблицами ширины w. Никакое креативное множество не может быть ограниченно таблично сводимым к простому множеству (П о с т [74], стр. 304).

34. Пусть α^n обозначает совокупность номеров n-ок с элементами из α. Показать, что α^n сводится к α таблицами ширины n. Существует такое простое множество α, что $\alpha^2 \not\leq_m \alpha$, $\alpha^3 \not\leq_m \alpha^2, \ldots$ ($\alpha \not\leq_m \beta$ означает, что α не m-сводимо к β) (Ф и ш е р [107]).

35. Согласно Шмульяну иммунное множество α называется *эффективно иммунным*, если существует такая частично рекурсивная функция $f(x)$, что для каждого значения x: если πx содержится в α, то $f(x)$ определено и число чисел в πx меньше $f(x)$. Рекурсивно перечислимое множество называется *эффективно простым*, если его дополнение эффективно иммунное. Показать, что построенное в п. 8.3 простое множество Поста эффективно простое. Каждое эффективно простое множество простое. Существуют простые множества, не являющиеся эффективно простыми (С а к с [92]).

36. Показать, что пересечение двух простых множеств есть простое множество, объединение двух простых множеств либо просто, либо имеет конечное дополнение, существуют два простых множества, объединение которых совпадает с всем множеством натуральных чисел. Те же утверждения имеют место и для гиперпростых множеств (Д е к к е р [28]).

§ 9. НУМЕРАЦИИ ПРОИЗВОЛЬНЫХ СОВОКУПНОСТЕЙ

В данном параграфе излагаются основы общей теории нумерованных совокупностей объектов. Одно простое свойство нумерации Клини, сформулированное в абстрактном виде, принимается в качестве аксиомы, определяющей так называемые полные нумерации, среди которых, помимо нумераций Клини и Поста, имеется и ряд других конкретных важных нумераций. Более детально изучаются свойства некоторых специальных нумераций семейств рекурсивно перечислимых множеств, что позволяет более отчетливо выделить место, занимаемое нумерациями Клини и Поста среди других возможных нумераций совокупностей частично рекурсивных функций и множеств. В конце параграфа рассматривается понятие рекурсивной неотделимости множеств, связанное с нумерациями и играющее заметную роль в теории рекурсивных функций.

9.1. Изоморфизм и эквивалентность нумераций. *Нумерацией* φ совокупности объектов $\boldsymbol{U}$ называется отображение некоторого множества $\boldsymbol{N}_\varphi$ натуральных чисел на совокупность $\boldsymbol{U}$. Число $n \in \boldsymbol{N}_\varphi$ называется φ-номером (или просто номером, если φ фиксировано) объекта $\varphi n \in \boldsymbol{U}$. Множество $\boldsymbol{N}_\varphi$ называется ***номерным множеством*** нумерации φ. Из этого определения следует, что каждый объект нумерованной совокупности имеет хотя бы один номер. Если каждый объект имеет в точности один номер, то нумерация называется одно-однозначной (или, короче, однозначной). Если номерное множество $\boldsymbol{N}_\varphi$ совпадает со всем натуральным рядом $\boldsymbol{N}$, то нумерация называется *простой*. Далее мы будем рассматривать лишь простые нумерации, хотя многие понятия и теоремы легко переносятся и на непростые нумерации (см. [52] и указанную там литературу). Соответственно этому далее под словом нумерация, если не оговорено противное, всегда понимается простая нумерация.

Пусть $\langle \boldsymbol{U}, \varphi \rangle$ — некоторая нумерованная совокупность $\boldsymbol{U}$ с нумерацией φ. Вводим на номерном множестве $\boldsymbol{N}_\varphi = \boldsymbol{N}$ бинарное отношение θ, полагая по определению

$$x\theta y \Longleftrightarrow \varphi x = \varphi y.$$

Ясно, что отношение θ рефлексивно, транзитивно и симметрично. Его иногда называют *нумерационной эквивалентностью* и вместо $x\theta y$ пишут $x \equiv y \pmod{\theta}$. Совокупности чисел, эквивалентных друг другу по mod θ, называются *нумерационными классами*. Иначе говоря, нумерационными классами называются совокупности всех номеров отдельных объектов из $\boldsymbol{U}$. Если n — число, то через θn обозначается множество чисел, сравнимых по mod θ с n, то есть θn есть множество всех номеров объекта φn. Система $\boldsymbol{U}/\theta$ всех нумерационных классов находится в естественном взаимно однозначном соответствии с совокупностью объектов $\boldsymbol{U}$. В каком-то смысле нумерацию φ можно считать вполне заданной, если задано отношение θ. С этой точки зрения теория нумерованных совокупностей — это теория рефлексивных, транзитивных и симметричных отношений, заданных на натуральном ряде.

Говорят, что общерекурсивная функция $f(x)$ *представляет гомоморфизм* нумерованного множества $\langle \boldsymbol{U}, \varphi \rangle$ на нумерованное множество $\langle \boldsymbol{V}, \psi \rangle$, если $f(x)$ отображает взаимно однозначно натуральный ряд на себя и если для каждых x, y

$$\varphi x = \varphi y \Rightarrow \psi f(x) = \psi f(y). \tag{1}$$

Из условия (1) следует, что отображение $h : \varphi x \to \psi f(x)$ есть однозначное (в одну сторону) отображение $\boldsymbol{U}$ на $\boldsymbol{V}$, которое, собственно, и называется гомоморфизмом, а функция $f(x)$ лишь «представляет» этот гомоморфизм.

Ясно, что если $f(x)$ представляет гомоморфизм нумерованного множества $\langle \boldsymbol{U}, \varphi \rangle$ на нумерованное множество $\langle \boldsymbol{V}, \psi \rangle$, а функция $g(x)$ представляет гомоморфизм $\langle \boldsymbol{V}, \psi \rangle$ на нумерованное множество $\langle \boldsymbol{W}, \chi \rangle$, то функция $g(f(x))$ представляет гомоморфизм $\langle \boldsymbol{U}, \varphi \rangle$ на $\langle \boldsymbol{W}, \chi \rangle$. Иначе говоря, *композиция гомоморфизмов есть гомоморфизм*.

Говорят, что общерекурсивная функция $f(x)$ *представляет изоморфизм* нумерованного множества $\langle \boldsymbol{U}, \varphi \rangle$ на нумерованное множество $\langle \boldsymbol{V}, \psi \rangle$, если $f(x)$ отображает взаимно однозначно натуральный ряд на себя и для каждых x, y

$$\varphi x = \varphi y \Leftrightarrow \psi f(x) = \psi f(y).$$

Таким образом, функция $f(x)$ представляет изоморфизм, если f представляет гомоморфизм $\langle \boldsymbol{U}, \varphi \rangle$ на $\langle \boldsymbol{V}, \psi \rangle$, а обратная функция f^{-1} представляет гомоморфизм $\langle \boldsymbol{V}, \psi \rangle$ на $\langle \boldsymbol{U}, \varphi \rangle$.

Нумерованные множества называются *изоморфными*, если существует функция, представляющая изоморфизм первого множества на второе. Из замечания о композиции гомоморфизмов следует, что отношение изоморфизма нумерованных множеств рефлексивно, симметрично и транзитивно.

Вводя понятия гомоморфизма и изоморфизма, мы рассматривали, вообще говоря, различные нумерованные совокупности объектов. Однако случай, когда сравниваются различные нумерации одного и того же множества, играет важную роль и требует новых понятий. Пусть φ, ψ — (простые) нумерации одной и той же совокупности объектов $\boldsymbol{U}$. Мы будем говорить, что нумерация φ *сводится к нумерации* ψ (символически $\varphi \leqslant \psi$), если существует общерекурсивная функция $f(x)$ такая, что

$$\varphi x = \psi f(x).$$

Нумерации φ и ψ называются *эквивалентными*, если каждая из них сводима к другой. С точки зрения теории алгоритмов эквивалентные нумерации — это такие нумерации, для которых существует алгоритм, позволяющий по произвольному номеру любого объекта из основной совокупности в одной нумерации найти некоторый номер того же объекта в другой нумерации.

Нумерация φ называется *рекурсивно изоморфной* нумерации ψ, если существует общерекурсивная функция $f(x)$, отображающая *взаимно однозначно* натуральный ряд $\boldsymbol{N}$ на себя и сводящая φ к ψ.

Значение понятия изоморфизма нумераций можно пояснить следующим образом. Вводя нумерацию φ совокупности $\boldsymbol{U}$, мы, естественно, начинаем смотреть на номер x объекта φx как на его координату в координатизации φ. Переход от нумерации φ к нумерации ψ означает, с этой точки зрения, переход от одной координатизации к другой. Какие же координатизации считать равноценными? С алгоритмической точки зрения наиболее «равноценными» следует считать изоморфные, так как в этом случае упомянутая выше функция $f(x)$ дает алгоритм, позволяющий по совокупности всех номеров объекта в одной нумерации находить не один номер, а все номера того же объекта в другой нумерации.

Мы ввели два понятия изоморфизма: изоморфизм двух нумерованных множеств и изоморфизм нумераций одного и того же множества. Следующий пример показывает, что эти понятия существенно различны.

Пусть семейство U состоит из всех натуральных чисел. Тождественное отображение U на U обозначим через φ. С другой стороны, пусть функция $g(x)$ осуществляет взаимно однозначное отображение натурального ряда на себя, но не является рекурсивной. Формулой $\psi x = g(x)$ вводим нумерацию ψ той же совокупности U. Нумерации φ и ψ как нумерации одного и того же множества не изоморфны. Действительно, изоморфизм φ и ψ равносилен существованию общерекурсивной функции $f(x)$, удовлетворяющей соотношению

$$\varphi x = \psi f(x).$$

Из определения нумераций φ, ψ следует, что $g(f(x)) = x$. Но это невозможно, так как g не рекурсивна, а f рекурсивна.

В то же время нумерованное множество $\langle U, \varphi \rangle$ изоморфно нумерованному множеству $\langle U, \psi \rangle$, так как заведомо выполняется соотношение $\varphi x = \varphi y \Longleftrightarrow \psi x = \psi y$.

Рассмотрим еще одну нумерацию того же множества U, определенную формулами

$$\chi(0) = 0, \quad \chi(x+1) = x.$$

Это значит, что объект 0 имеет в нумерации χ два номера 0 и 1, а все остальные объекты имеют по одному номеру. Нумерации φ, χ эквивалентны. Однако они не изоморфны, ибо соответствующие нумерационные классы у изоморфных нумераций имеют одинаковое количество чисел, тогда как все классы нумерации φ имеют по одному числу, а нумерация χ имеет класс, состоящий из двух элементов.

Следующий способ образования из одного нумерованного множества другого часто оказывается полезным. Пусть задана совокупность объектов U, снабженная нумерацией φ. Определим на U какое-нибудь отношение эквивалентности ϱ и рассмотрим систему U/ϱ всех классов друг другу ϱ-эквивалентных объектов из U. Пусть ϱa — класс всех объектов из U, ϱ-эквивалентных какому-либо объекту a. Вводим нумерацию φ_ϱ системы U/ϱ, полагая

для произвольного $x \in N$

$$\varphi_\rho x = \rho\varphi x. \tag{2}$$

Из этой формулы непосредственно видно, что тождественная функция $f(x) = x$ *представляет гомоморфизм* нумерованного множества $\langle U, \varphi\rangle$ на нумерованное множество $\langle U/\rho, \varphi_\rho\rangle$. Множество U/ρ называется *фактор-множеством* от U по эквивалентности ρ, а нумерация φ_ρ называется *фактор-нумерацией* от φ по ρ. Из той же формулы (2) видно, что нумерационные классы чисел для фактор-нумерации φ_ρ являются объединениями соответствующих нумерационных классов для исходной нумерации φ.

Теорема 1 (о гомоморфизме). *Пусть функция $f(x)$ представляет гомоморфизм нумерованного множества $\langle U, \varphi\rangle$ на нумерованное множество $\langle V, \psi\rangle$. Вводим на совокупности U эквивалентность ρ, называя объекты из U ρ-эквивалентными, если их гомоморфные образы в V совпадают, то есть полагая*

$$(\varphi x)\,\rho\,(\varphi y) \Leftrightarrow \psi f(x) = \psi f(y). \tag{3}$$

Тогда функция $f(x)$ представляет изоморфизм фактор-множества $\langle U/\rho, \varphi_\rho\rangle$ на $\langle V, \psi\rangle$.

В самом деле, из (3) следует

$$\psi f(x) = \psi f(y) \Rightarrow \varphi_\rho x = \varphi_\rho y.$$

Обратная импликация также очевидна.

Образец гомоморфизма, представляемого функцией $f(x) = x$, дают нумерации Клини $\varkappa$ и Поста π, так как заведомо

$$\varkappa m = \varkappa n \Rightarrow \pi m = \pi n.$$

Упоминающееся в теореме 1 отношение ρ здесь означает: две функции ρ-эквивалентны, если совокупности значений этих функций равны.

9.2. Односводимость нумераций. Нумерация φ множества U называется *односводимой* к нумерации ψ того же множества, если существует общерекурсивная функция $f(x)$, принимающая *различные* значения при различных значениях аргумента и сводящая φ к ψ. Если функция $f(x)$ односводит нумерацию φ к нумерации ψ, а функция $g(x)$ односводит ψ к нумерации χ, то функция $g(f(x))$, очевидно, односводит φ к χ.

Нумерации φ и ψ называются *одноэквивалентными*, если каждая из них односводима к другой.

Понятие односводимости представляет собой специализацию понятия сводимости. Поэтому из односводимости φ к ψ или одноэквивалентности φ и ψ вытекает соответственно сводимость или эквивалентность φ, ψ. Обратное не всегда верно. Однако существуют простые условия, при которых верно и обратное. Для формулировки этих условий введем еще одно понятие.

Нумерацию ψ совокупности U назовем *нумерацией с эффективно бесконечными классами*, если существует двуместная общерекурсивная функция $B(x, y)$, обладающая следующими свойствами:

а) $\psi B(x, y) = \psi x$;
б) $y \neq z \Rightarrow B(x, y) \neq B(x, z)$.

Функция $B(x, y)$ доставляет нам алгоритм, который по одному какому-нибудь номеру x объекта позволяет найти бесконечное число различных номеров $B(x, 0)$, $B(x, 1)$, ... того же объекта.

Теорема 1. *Если нумерация φ сводима к нумерации ψ, имеющей эффективно бесконечные классы, то φ односводима к ψ.*

Пусть функция $f(x)$ сводит φ к ψ, и пусть функция $B(x, y)$ обладает свойствами а), б) относительно нумерации ψ. Функцию $g(x)$, односводящую φ к ψ, можно построить следующим образом.

Полагаем по определению $g(0) = f(0)$ и далее, считая значения $g(0)$, $g(1)$, ..., $g(n)$ уже известными, полагаем

$$h(n) = \mu_t (B(f(n+1), t) \notin \{g(0), \ldots, g(n)\}), \qquad (1)$$

$$g(n+1) = B(f(n+1), h(n)). \qquad (2)$$

Из (1) следует, что $g(n+1) \notin \{g(0), \ldots, g(n)\}$, то есть что функция $g(x)$ принимает различные значения при различных значениях аргумента, а из (2) и свойств функции $B(x, y)$ имеем

$$\psi g(n) = \psi f(n) = \varphi n.$$

Остается проверить, что функция g общерекурсивная. С точки зрения теории алгоритмов это ясно, так как для

вычисления значений функции g указан регулярный процесс. Однако для строгого доказательства этого факта надо показать, что функция g получается операторами подстановки, примитивной рекурсии и μ-оператором из функций B, f и других общерекурсивных функций. Необходимые вычисления не представляют особой трудности и мы их здесь опустим (см. задачи 1, 2 к этому параграфу).

Если внимательно просмотреть доказательство теоремы 1, то легко обнаружить, что оно показывает справедливость не только теоремы 1, но и следующего, несколько более общего утверждения.

Теорема 1а. *Пусть нумерация φ сводима к нумерации ψ. Если при этом существует общерекурсивная функция $B_0(x, y)$, удовлетворяющая требованиям*

а) $\psi B_0(x, y) = \varphi x$;
б) $y \neq z \Rightarrow B_0(x, y) \neq B_0(x, z)$,

то нумерация φ односводима к нумерации ψ.

Действительно, функцию $g(x)$, односводящую φ к ψ, можно определить по тем же формулам (1), (2), заменив в них лишь подформулы $B(f(n+1), t)$ и $B(f(n+1), h(n))$ соответственно выражениями $B_0(n+1, t)$ и $B_0(n+1, h(n))$.

Ясно, что теорема 1 есть частный случай теоремы 1а: если функция $B(x, y)$ для нумерации ψ известна и нумерация φ сводится функцией $f(x)$ к нумерации ψ, то функция

$$B_0(x, y) = B(f(x), y)$$

обладает свойствами а), б) из теоремы 1а.

Мы теперь сформулируем и докажем одну из центральных теорем теории нумерованных множеств, имеющую большое значение как для самой этой теории, так и для ее приложений.

Теорема 2 (об изоморфизме нумераций). *Одноэквивалентные нумерации произвольной совокупности объектов изоморфны между собой.*

Пусть φ, ψ — одноэквивалентные нумерации какой-нибудь совокупности объектов U. Обозначим через $f(x)$ функцию, односводящую нумерацию φ к нумерации ψ, и через $g(x)$ функцию, односводящую ψ к φ. Нам надо

указать алгоритм для вычисления значений новой функции $h(x)$, которая представляла бы изоморфизм φ на ψ. Для этого придется ввести несколько вспомогательных понятий и функций.

Финитным соответствием условимся называть каждую конечную последовательность

$$\langle x_0, y_0\rangle, \langle x_1, y_1\rangle, \ldots, \langle x_k, y_k\rangle \tag{3}$$

пар натуральных чисел, удовлетворяющую условиям

$$\begin{aligned} x_i = x_j &\Longleftrightarrow y_i = y_j, \\ \varphi x_i = \psi y_i &\qquad (i, j = 0, 1, \ldots, k). \end{aligned} \tag{4}$$

Номером финитного соответствия (3) назовем число

$$n = \prod_{i=0}^{k} p_i^{1+[x_i, y_i]},$$

где $p_0 = 2$, $p_1 = 3$, ... — последовательные простые числа, а $[x, y]$ — номер пары $\langle x, y\rangle$, вычисляемый согласно п. 7.1.

С помощью заданных функций f, g мы хотим теперь построить общерекурсивные функции $S(m, n)$, $T(m, n)$, значения которых были бы равны номерам финитных соответствий

$$\langle x_0, y_0\rangle, \langle x_1, y_1\rangle, \ldots, \langle x_k, y_k\rangle, \langle m, y_{k+1}\rangle \tag{5}$$

и соответственно

$$\langle x_0, y_0\rangle, \langle x_1, y_1\rangle, \ldots, \langle x_k, y_k\rangle, \langle x_{k+1}, m\rangle, \tag{6}$$

где n — номер соответствия (3), m — произвольное число, а x_{k+1}, y_{k+1} — подходящие числа, зависящие от m, n. Соответствия (5) и (6) представляют собой расширения соответствия (3), которые вводят в игру произвольно заданное число m. Возможность таких расширений и доказывается посредством построений функций $S(m, n)$ и $T(m, n)$.

Укажем сначала алгоритм для вычисления $S(m, n)$. Пусть заданы число m и некоторое соответствие (3), номер которого есть n. Нам надо указать, какое число мы выбираем в качестве y_{k+1}. Этот выбор производится по-разному в каждом из следующих трех возможных случаев.

а) Если $m \in \{x_0, \ldots, x_k\}$ и i — наименьшее число, удовлетворяющее требованию $x_i = m$, то полагаем $y_{k+1} = y_i$.

б) Если $m \notin \{x_0, \ldots, x_k\}$ и $f(m) \notin \{y_0, \ldots, y_k\}$, то полагаем $y_{k+1} = f(m)$.

в) Если $m \notin \{x_0, \ldots, x_k\}$, но $f(m) \in \{y_0, \ldots, y_k\}$, то число y_{k+1} строим посредством следующего процесса. Берем наименьшее число i_1, для которого $f(m) = y_{i_1}$, и рассматриваем $f(x_{i_1})$. Если $f(x_{i_1}) \in \{y_0, \ldots, y_k\}$, то берем наименьшее i_2, для которого $f(x_{i_1}) = y_{i_2}$, и рассматриваем $f(x_{i_2})$. Продолжая этот процесс, получим серию чисел $y_{i_1}, y_{i_2}, \ldots$ Мы хотим показать, что эта серия оборвется, то есть что найдется число s, для которого $f(x_{i_s}) \notin \{y_0, \ldots, y_k\}$.

Докажем, что числа $y_{i_1}, y_{i_2}, \ldots$ различны. Пусть для некоторого u числа $y_{i_1}, \ldots, y_{i_u}$ попарно различны и $f(x_{i_u}) = y_{i_{u+1}}$. Если бы оказалось, что $y_{i_{u+1}} = y_{i_v}$ $(1 \leqslant v \leqslant u)$, то мы имели бы $f(x_{i_u}) = f(x_{i_{v-1}})$. Так как функция $f(x)$ при разных значениях аргумента принимает разные значения, то из последнего равенства следует $x_{i_u} = x_{i_{v-1}}$ и, значит, $y_{i_u} = y_{i_{v-1}}$, вопреки предположению.

Итак, все члены последовательности $y_{i_1}, y_{i_2}, \ldots$ различны и все они содержатся в множестве $\{y_0, \ldots, y_k\}$. Поэтому последовательность будет иметь последний элемент y_{i_s} и, таким образом,

$$f(m) = y_{i_1},\ f(x_{i_1}) = y_{i_2},\ \ldots,\ f(x_{i_{s-1}}) = y_{i_s},$$
$$f(x_{i_s}) \notin \{y_0, \ldots, y_k\}. \qquad (7)$$

По определению полагаем $y_{k+1} = f(x_{i_s})$.

Пара $\langle m, y_{k+1}\rangle$ построена. Надо показать, что последовательность (5) представляет собой финитное соответствие при условии, что финитным соответствием является последовательность (3). В случаях а) и б) это очевидно. В случае в) второе из требований (4) выполняется автоматически, так как ввиду цепочки равенств (7) последовательно получаем

$$\varphi m = \psi(f(m)) = \psi y_{i_1} = \varphi x_{i_1} = \psi f(x_{i_1}) = \ldots = \psi f(x_{i_s}) = \psi y_{k+1}.$$

Первое из условий (4) в случае в) также выполнено, так как для $i < k + 1$ имеем $x_i \neq m$, а из $y_i = y_{k+1}$ следовало бы соотношение $f(x_{i_s}) = y_i \in \{y_0, \ldots, y_k\}$, невозможное ввиду (7).

Меняя ролями $f, x_0, \ldots, x_k$ и $g, y_0, \ldots, y_k$, аналогичным путем по заданному соответствию (3) и числу m строим

расширенное соответствие (6). Как уже говорилось, номера соответствий (5) и (6) обозначаются через $S(m, n)$ и $T(m, n)$, где n — номер соответствия (3).

Конечно, нас интересуют лишь случаи, когда последовательность (3) является финитным соответствием. Однако с формальной точки зрения описанные выше процессы нахождения числа y_{k+1} в случаях а), б) дают определенный результат и тогда, когда последовательность (3) не есть финитное соответствие. Чтобы достичь этого в случае в), достаточно считать серию $y_{i_1}, y_{i_2}, \ldots, y_{i_s}$ обрывающейся, как только окажется, что $f(x_{i_s}) \notin \{y_0, \ldots, y_k\}$ или что y_{i_s} совпадает с каким-либо предыдущим членом в последовательности $y_{i_1}, y_{i_2}, \ldots$

После этого дополнительного замечания мы можем считать функции $S(m, n)$, $T(m, n)$ определенными для всех значений m, n, исключая $n = 0, 1$, так как последовательностей с такими номерами вообще нет. Но мы и для $n = 0, 1$ доопределим функции $S(m, n)$, $T(m, n)$ каким-либо регулярным способом. В результате будем иметь всюду определенные функции S, T, значения которых вычисляются при помощи точного алгоритма. Согласно тезису Чёрча отсюда следует рекурсивность функций S, T. Строгое доказательство рекурсивности S, T требует, чтобы функции S, T были выражены через f, g и другие рекурсивные функции при помощи операций подстановки и μ-операций. Ввиду рутинного характера соответствующих вычислений мы здесь их приводить не будем.

Итак, нами построены общерекурсивные функции S и T. Функция S позволяет финитное соответствие с номером n пополнить парой $\langle m, y_{k+1}\rangle$, содержащей в качестве *левого* элемента произвольно заданное число m, а функция T позволяет финитное соответствие с номером n пополнить парой $\langle x_{k+1}, m\rangle$, содержащей произвольно заданное число m в качестве *правого* элемента.

Берем теперь простейшее соответствие, состоящее всего из одной пары $\langle 0, f(0)\rangle$, и называем его *начальным*, а затем постепенно расширяем его, применяя поочередно функции S и T так, чтобы в целом получилось отображение всего натурального ряда на себя. Для более точного описания этого процесса вводим функцию $w(x)$, значение которой $w(n)$ будет равно номеру n-го расширения начального

соответствия. Полагаем по определению

$$w(0) = p_0^{1+[0,\ f(0)]},$$

$$w(n+1) = \begin{cases} T\left(\frac{n}{2},\ w(n)\right), & \text{если } n \text{ четно}, \\ S\left(\frac{n+1}{2},\ w(n)\right), & \text{если } n \text{ нечетно}. \end{cases} \tag{8}$$

Отсюда видно, что функция $w(x)$ возникает примитивной рекурсией из функции

$$S\left(\left[\frac{n+1}{2}\right],\ z\right)\left(n-2\left[\frac{n}{2}\right]\right)+ \\ +T\left(\left[\frac{n}{2}\right],\ z\right)\left|n-2\left[\frac{n}{2}\right]-1\right|$$

и потому $w(x)$ — общерекурсивная функция.

Рассмотрим, наконец, функцию $y = h(x)$, значение которой y равно правому числу той пары финитного соответствия с номером $w(2x)$, левое число которой равно x.

Согласно схеме (8) последняя пара финитного соответствия с номером $w(2x)$ имеет вид $\langle x,\ y_{k+1}\rangle$, причем

$$y_{k+1} = [\mathrm{ex}(2x,\ w(2x)-1)]_{22},$$

где $[z]_{22}$ — правый элемент пары с номером z (п. 7.1). Поэтому

$$y = h(x) = [\mathrm{ex}(2x,\ w(2x)-1)]_{22}$$

и, следовательно, функция $h(x)$ общерекурсивная. Если $x_1 < x$, то соответствие с номером $w(2x)$ содержит пары $\langle x_1,\ h(x_1)\rangle$ и $\langle x,\ h(x)\rangle$. Из (4) получаем $h(x_1) \neq h(x)$. С другой стороны, соответствие с номером $2m+1$ согласно (8) оканчивается парой вида $\langle x_{k+1},\ m\rangle$ и, значит, уравнение $h(x) = m$ имеет решение x_{k+1} для каждого m. Таким образом, функция $h(x)$ отображает взаимно однозначно натуральный ряд на себя, удовлетворяет требованию $\varphi x = \psi h(x)$ и представляет искомый изоморфизм нумерации φ на нумерацию ψ.

Доказательство теоремы 2 завершено.

Теоремы 1 и 2 относятся к нумерациям одной и той же совокупности объектов. Однако нетрудно получить соответственные теоремы и для нумераций различных совокуп-

ностей объектов. Мы ограничимся здесь переформулировкой теоремы 2.

Т е о р е м а 3. *Пусть заданы какие-либо нумерованные множества* $\langle \boldsymbol{M}_1, \varphi\rangle$, $\langle \boldsymbol{M}_2, \psi\rangle$ *и взаимно однозначное отображение* ζ *совокупности объектов* $\boldsymbol{M}_1$ *первого множества на совокупность объектов второго. Если существуют общерекурсивные функции* $f(x)$, $g(x)$, *принимающие в различных точках различные значения и удовлетворяющие требованиям*

$$\zeta(\varphi x) = \psi f(x), \qquad \zeta(\varphi g(x)) = \psi x, \tag{9}$$

то отображение ζ *есть изоморфизм* $\langle \boldsymbol{M}_1, \varphi\rangle$ *на* $\langle \boldsymbol{M}_2, \psi\rangle$, *то есть существует общерекурсивная функция* $h(x)$, *взаимно однозначно отображающая натуральный ряд на себя и удовлетворяющая требованию*

$$\zeta(\varphi x) = \psi h(x).$$

Действительно, введем еще одну нумерацию ψ_1 совокупности $\boldsymbol{M}_2$, полагая

$$\psi_1 n = \zeta(\varphi n). \tag{10}$$

Соотношения (9) показывают, что функция $f(x)$ односводит ψ_1 к ψ, а функция $g(x)$ односводит ψ к ψ_1. По теореме 2 отсюда заключаем, что нумерации ψ и ψ_1 рекурсивно изоморфны, то есть что существует общерекурсивная функция $h(x)$, взаимно однозначно отображающая натуральный ряд на себя и удовлетворяющая требованию $\psi_1 x = \psi h(x)$. В силу (9) и (10) это соотношение можно переписать в форме $\zeta(\varphi x) = \psi h(x)$, что и требовалось.

Чтобы проиллюстрировать действенность теоремы 3, докажем, например, что *нумерация Клини* $\varkappa$ *всех одноместных частично рекурсивных функций и нумерация Клини* $\varkappa_2$ *всех двуместных частично рекурсивных функций изоморфны.*

Обозначим через ζ отображение, при котором одноместной частично рекурсивной функции $\boldsymbol{K}(n, x)$ ставится в соответствие двуместная частично рекурсивная функция $\boldsymbol{K}(n, [x, y])$. Ясно, что ζ взаимно однозначно отображает совокупность $\mathfrak{F}^1_{\text{p. r}}$ всех одноместных частично рекурсивных функций на совокупность $\mathfrak{F}^2_{\text{p. r}}$ всех двуместных частично рекурсивных функций. Ищем такое постоянное a, чтобы

$$\boldsymbol{K}(n, [x, y]) = \boldsymbol{K}([a, n], x, y).$$

Отсюда получаем $\zeta\varkappa n = \varkappa_2 [a, n]$. Следовательно, функция $f(x) = [a, x]$ «односводит» $\varkappa$ к $\varkappa_2$. С другой стороны, ищем такую постоянную b, чтобы

$$\boldsymbol{K}(n, [t]_{21}, [t]_{22}) = \boldsymbol{K}(b, n, t).$$

Подставляя сюда вместо t выражение $[x, y]$, получим

$$\boldsymbol{K}([b, n], [x, y]) = \boldsymbol{K}(n, x, y),$$

то есть $\zeta(\varkappa[b, n]) = \varkappa_2 n$. Это показывает, что функция $g(x) = [b, x]$ «односводит» нумерацию $\varkappa_2$ к нумерации $\varkappa$. Слово «односводит» взято нами в кавычки, так как понятие односводимости было нами определено лишь для различных нумераций одной и той же совокупности, а здесь мы имеем дело с разными совокупностями и «односводимость» следует понимать в смысле, указанном в теореме 3. Пользуясь этой теоремой, мы заключаем, что отображение ζ есть изоморфизм $\langle \mathfrak{F}^1_{\text{р. г}}, \varkappa \rangle$ на $\langle \mathfrak{F}^2_{\text{р. г}}, \varkappa_2 \rangle$.

9.3. Полные нумерации. Нумерация φ совокупности объектов $\boldsymbol{M}$ называется *полной* (см. [52]), если в $\boldsymbol{M}$ существует *особый* объект о, обладающий следующим свойством: для каждой частично рекурсивной функции $F(x)$ существует такая общерекурсивная функция $G(x)$, что для каждого числа x

$$\varphi G(x) = \begin{cases} \varphi F(x), & \text{если } F(x) \text{ определено}, \\ \text{о}, & \text{если } F(x) \text{ не определено}. \end{cases} \quad (1)$$

Здесь, как и всюду, под нумерацией мы понимаем простую нумерацию, у которой номерное множество совпадает со всем натуральным рядом. Если совокупность $\boldsymbol{M}$ содержит лишь один объект, то простых нумераций у $\boldsymbol{M}$ лишь одна и она, очевидно, полная.

Теорема 1. *Гомоморфный образ совокупности с полной нумерацией есть совокупность с полной нумерацией.*

Действительно, пусть функция $h(x)$ представляет гомоморфизм полно нумерованной совокупности $\langle \boldsymbol{M}, \varphi \rangle$ с особым объектом о на нумерованную совокупность $\langle \boldsymbol{M}_1, \psi \rangle$ и, значит,

$$\varphi x = \varphi y \Rightarrow \psi h(x) = \psi h(y). \quad (2)$$

Пусть a — какой-нибудь φ-номер о. Положим $\text{о}_1 = \psi h(a)$ и рассмотрим произвольную частично рекурсивную функ-

цию $F(x)$. По условию найдется такая общерекурсивная функция $G(x)$, для которой

$$\varphi G(x) = \varphi h^{-1}(F(x)), \quad \text{если } F(x) \text{ определено,}$$
$$\varphi G(x) = \varphi a, \quad \text{если } F(x) \text{ не определено.}$$

В силу (2) мы можем в этих равенствах символ φ заменить символом ψh и получить, таким образом, условия полноты нумерации ψ с особым элементом o_1.

Из теоремы 1, в частности, следует, что если нумерованное множество изоморфно множеству с полной нумерацией, то оно само имеет полную нумерацию.

Теорема 2. *Нумерация Клини полная, имеющая нигде не определенную функцию своим особым элементом.*

Пусть $F(x)$ — какая-нибудь частично рекурсивная функция. Ищем такое число a, чтобы для всех значений x, t было

$$\boldsymbol{K}(F(x), t) = \boldsymbol{K}([a, x], t).$$

Таким образом, если $F(x)$ определено, то функция номера $[a, x]$ совпадает с функцией номера $F(x)$. Если же $F(x)$ не определено, то функция номера $[a, x]$ есть нигде не определенная функция o. Следовательно, общерекурсивная функция $G(x) = [a, x]$ удовлетворяет условию (1) и нумерация $\varkappa$ полная.

В п. 9.2 было установлено, что нумерация Клини $\varkappa_s$ всех s-местных частично рекурсивных функций изоморфна нумерации $\varkappa$. Поэтому вместе с нумерацией $\varkappa$ полными являются и все нумерации $\varkappa_s$. В п. 9.1 показано, что нумерация Поста $\boldsymbol{\pi}$ есть гомоморфный образ нумерации $\varkappa$. Следовательно, $\boldsymbol{\pi}$ — также полная нумерация, причем особым элементом для $\boldsymbol{\pi}$ является пустое множество.

Сопоставляя теоремы 1 и 2, видим, что, разбивая одноместные частично рекурсивные функции совершенно произвольным образом на классы и беря соответствующую фактор-нумерацию от $\varkappa$, мы получим полную нумерацию. Это показывает, что запас полных нумераций достаточно большой.

Мы хотим теперь некоторые свойства нумерации Клини, полученные в § 7, распространить на произвольные полные нумерации.

Теорема 3. *Для каждой полной нумерации* φ *и каждого* r *существует общерекурсивная функция* $P_{\varphi}^{r}(x_1, \ldots, x_r)$, *обладающая следующим свойством: для каждой частично рекурсивной* r-*местной функции* $H(x_1, \ldots, x_r)$ *существует такое число* a, *что для любых значений* $x_2, \ldots, x_r$

$$\varphi P_{\varphi}^{r}(a, x_2, \ldots, x_r) = \begin{cases} \varphi H(a, x_2, \ldots, x_r), & \text{если } H(a, x_2, \ldots, x_r) \text{ определено}, \\ o & \text{в противном случае}. \end{cases} \quad (3)$$

Рассмотрим вспомогательную функцию

$$F(x) = \boldsymbol{K}([x]_{r1}, \ldots, [x]_{rr}),$$

где $[x]_{ij}$ — введенные в п. 7.1 нумерационные функции. Вследствие полноты φ существует общерекурсивная функция $G(x)$, связанная с $F(x)$ соотношениями (1). Из (1) подстановкой $x = [x_1, \ldots, x_r]$ получаем соотношение

$$\varphi G([x_1, \ldots, x_r]) = \begin{cases} \varphi \boldsymbol{K}(x_1, \ldots, x_r), & \text{если } \boldsymbol{K}(x_1, \ldots, x_r) \text{ определено}, \\ o & \text{в противном случае}. \end{cases} \quad (4)$$

Покажем, что функция

$$P_{\varphi}^{r}(x_1, \ldots, x_r) = G([x_1, \ldots, x_r])$$

удовлетворяет требованиям теоремы 3. Так как функция $\boldsymbol{K}$ универсальная, то при некотором c получим тождество

$$H([x_1, x_1], x_2, \ldots, x_r) = \boldsymbol{K}([c, x_1], x_2, \ldots, x_r).$$

Полагая здесь $x_1 = c$, $a = [c, c]$ и подставляя в (4) вместо x_1 число a, получим (3).

Теорема 4. *Для каждой полной нумерации* φ *и каждой частично рекурсивной функции* $f(x_1, \ldots, x_r)$ *существует такая общерекурсивная функция* $g(x_2, \ldots, x_r)$, *что*

$$\varphi g(x_2, \ldots, x_r) = \varphi f(g(x_2, \ldots, x_r), x_2, \ldots, x_r),$$

если $f(g(x_2, \ldots, x_r), x_2, \ldots, x_r)$ *определено, и*

$$\varphi g(x_2, \ldots, x_r) = o \text{ в противном случае}.$$

Рассмотрим функцию

$$H(x_1, \ldots, x_r) = f(P_\varphi^r(x_1, \ldots, x_r), x_2, \ldots, x_r),$$

где функция P_φ^r указана в теореме 3. Согласно теореме 3 найдется число a такое, что

$$\varphi P_\varphi^r(a, x_2, \ldots, x_r) = \varphi f(P_\varphi^r(a, \ldots, x_r), x_2, \ldots, x_r),$$

если выражение, стоящее справа, определено, и

$$\varphi P_\varphi^r(a, x_2, \ldots, x_r) = \mathrm{o}$$

в противном случае. Отсюда видно: чтобы удовлетворить теореме 4, достаточно в качестве функции $g(x_2, \ldots, x_r)$ взять $P_\varphi^r(a, x_2, \ldots, x_r)$.

Основное значение далее будет иметь не сама теорема 4, а ее частный случай:

Следствие 1. *Пусть φ — полная нумерация. Тогда для каждой общерекурсивной функции $f(x)$ уравнение*

$$\varphi f(x) = \varphi x \tag{5}$$

имеет решение («неподвижную» точку) x. При этом существует алгоритм, позволяющий по клиниевскому номеру f находить определенное решение x уравнения (5).

Иначе говоря, существует общерекурсивная функция $g(n)$ такая, что

$$\varphi \boldsymbol{K}(n, g(n)) = \varphi g(n) \tag{6}$$

для всех тех значений n, для которых функция $\boldsymbol{K}(n, t)$ всюду определенная. Но существование функции $g(n)$ с этими свойствами утверждает теорема 4, если в ее условии положить $f(x_1, x_2) = \boldsymbol{K}(x_2, x_1)$.

Нижеследующая важная теорема была сначала доказана Роджерсом [91] для нумерации Клини и позже была распространена на произвольные полные нумерации ([51]).

Теорема 5. *Если полная нумерация φ совокупности $\boldsymbol{U}$ сводится к какой-нибудь ее нумерации ψ, то φ односводится к ψ. Каждая нумерация, эквивалентная полной нумерации, изоморфна последней, а потому и полна.*

Второе утверждение теоремы вытекает из первого, так как одноэквивалентные нумерации в силу теоремы 2 из п. 9.2 являются изоморфными. Поэтому докажем лишь первое утверждение.

Пусть общерекурсивная функция $f(x)$ сводит φ к ψ. Если совокупность U имеет лишь один объект, то утверждение теоремы 5 тривиально. Поэтому мы можем предполагать, что существуют числа a, b, являющиеся φ-номерами различных объектов из U. Нам надо построить общерекурсивную функцию $B_0(x, y)$, обладающую следующими свойствами (ср. п. 9.2):

$$\psi B_0(x, y) = \varphi x, \qquad y \neq z \Rightarrow B_0(x, y) \neq B_0(x, z).$$

Строим сначала вспомогательную общерекурсивную функцию $S(x, y)$, удовлетворяющую требованиям

$$\varphi S(x, y) = \varphi x, \ f(S(x, y)) \neq f(S(x, z)) \qquad (y \neq z). \quad (7)$$

Полагаем по определению $S(x, 0) = x$ и далее по индукции допускаем, что для некоторого r значения $S(x, 0)$, $S(x, 1)$, ..., $S(x, r)$ уже найдены и притом такие, что требования (7) для $y, z = 0, \ldots, r$ выполнены. Вводим одноместные частичные функции

$$h_1(t) = \begin{cases} x, & \text{если } f(t) \in \{f(S(x, 0)), \ldots, f(S(x, r))\}, \\ a & \text{в противном случае;} \end{cases} \quad (8)$$

$$h_2(t) = \begin{cases} b, & \text{если } f(t) \in \{f(S(x, 0)), \ldots, f(S(x, r))\}, \\ a & \text{в противном случае.} \end{cases} \quad (9)$$

Функции h_1, h_2 частично рекурсивные. Более того, существует алгоритм, позволяющий по числам x, $S(x, 0)$, ..., $S(x, r)$ находить определенные клиниевские номера n_1, n_2 функций h_1, h_2. Полагаем, далее, $m_1 = g(n_1)$, $m_2 = g(n_2)$, где $g(t)$ — функция из соотношения (6). Следовательно,

$$\varphi h_1(m_1) = \varphi m_1, \ \varphi h_2(m_2) = \varphi m_2. \quad (10)$$

Если $f(m_1) \notin \{f(S(x, 0)), \ldots, f(S(x, r))\}$, то полагаем $S(x, r+1) = m_1$. Из соотношений (8), (10) в этом случае имеем $\varphi m_1 = \varphi x$ и потому требование (7) выполняется для $y, z = 0, \ldots, r, r+1$.

Если $f(m_1) \in \{f(S(x, 0)), \ldots, f(S(x, r))\}$, то полагаем $S(x, r+1) = m_2$. Покажем, что и в этом случае требование (7) выполнено для $y, z = 0, \ldots, r+1$. По предположению для некоторого i, $0 \leqslant i \leqslant r$, имеем $f(m_1) =$

$=f(S(x, i))$ и, значит, $\varphi m_1=\varphi(S(x, i))=\varphi x=\varphi h_1(m_1)$. Но в рассматриваемом случае согласно (8) $h_1(m_1)=a$. Поэтому $\varphi m_1=\varphi x=\varphi a$.

Если бы оказалось, что $f(m_2)\in\{f(S(x, 0)), \ldots \ldots, f(S(x, r))\}$, то аналогичным путем получились бы равенства $\varphi m_2=\varphi x=\varphi b$, противоречащие уже установленным соотношениям $\varphi x=\varphi a\neq\varphi b$. Поэтому $f(m_2)\notin \notin\{f(S(x, 0)), \ldots, f(S(x, r))\}$ и в силу (9) $h_2(m_2)=a$.

Отсюда имеем

$$f(S(x, r+1))\notin\{f(S(x, 0)), \ldots, f(S(x, r))\},$$

$$\varphi S(x, r+1)=\varphi m_2=\varphi a=\varphi x.$$

Таким образом, требования (7) в рассматриваемом случае выполнены и функцию $S(x, y)$ можно считать построенной.

Из регулярности процесса построения функции $S(x, y)$ видна ее общерекурсивность. Формальное доказательство этого предоставляется читателю. Свойства (7) показывают, что в качестве искомой функции $B_0(x, y)$ для нумерации ψ может быть взята функция $f(S(x, y))$, и доказательство теоремы 5 закончено.

9.4. Семейства объектов нумерованных совокупностей. Нумерация элементов произвольной совокупности $\boldsymbol{U}$ позволяет основные понятия теории числовых рекурсивных множеств перенести и на произвольные совокупности $\boldsymbol{U}$. Однако перенесение это можно делать различными способами, в силу чего понятия рекурсивности, рекурсивной перечислимости и т. п. при переходе от множеств чисел к семействам объектов разветвляются на *вполне* рекурсивные, *слабо* рекурсивные и т. д. Мы рассмотрим пока лишь понятия с приставкой «вполне»: вполне рекурсивные, вполне перечислимые и т. д.

Пусть $\boldsymbol{U}$ — совокупность объектов с нумерацией φ. Семейство $\boldsymbol{S}$ объектов из $\boldsymbol{U}$ будем называть нетривиальным, если $\boldsymbol{S}$ непусто и не совпадает с $\boldsymbol{U}$. Символом $\varphi^{-1}\boldsymbol{S}$ будем обозначать множество *всех* φ-номеров *всех* объектов из $\boldsymbol{S}$. Если $\boldsymbol{A}$ — некоторое множество чисел, то через $\varphi\boldsymbol{A}$ будет обозначаться φ-образ $\boldsymbol{A}$ в $\boldsymbol{U}$, то есть $\varphi\boldsymbol{A}$ есть семейство объектов, хотя бы один φ-номер которых содержится в $\boldsymbol{A}$.

Семейство $\boldsymbol{S}$ называется *вполне перечислимым* (*вполне рекурсивным*, *вполне креативным*), если множество $\varphi^{-1}\boldsymbol{S}$

рекурсивно перечислимое (рекурсивное, креативное) (ср. п. 7.2).

Следующая теорема является дословным распространением теоремы Райса, доказанной в п. 7.2 для нумерации Клини, на произвольные полные нумерации.

Теорема 1. *Вполне рекурсивные семейства объектов в совокупностях с полными нумерациями тривиальны.*

В самом деле, допустим, напротив, что совокупность $\boldsymbol{U}$ с полной нумерацией φ имеет нетривиальное вполне рекурсивное семейство объектов $\boldsymbol{S}$. Это означает, что множество $\varphi^{-1}\boldsymbol{S}$ не пусто, отлично от натурального ряда и рекурсивно. Но тогда непустым и рекурсивным будет и дополнение $\varphi^{-1}\boldsymbol{S}'$ множества $\varphi^{-1}\boldsymbol{S}$. Пусть $a \in \varphi^{-1}\boldsymbol{S}$, $b \in \varphi^{-1}\boldsymbol{S}'$. Строим вспомогательную функцию

$$f(x) = \begin{cases} a, & x \in \varphi^{-1}\boldsymbol{S}', \\ b, & x \in \varphi^{-1}\boldsymbol{S}. \end{cases}$$

Эта функция общерекурсивная (п. 6.3). Согласно теореме о неподвижной точке (п. 9.3) существует число c, для которого $\varphi f(c) = \varphi c$. Как и в п. 7.2, легко убеждаемся, что оба предположения: $c \in \varphi^{-1}\boldsymbol{S}$, $c \in \varphi^{-1}\boldsymbol{S}'$ ведут к противоречию.

Теорема 2. *Никакое нетривиальное вполне перечислимое семейство $\boldsymbol{S}$ объектов совокупности $\boldsymbol{U}$, имеющей полную нумерацию φ, не может содержать особого элемента* о *нумерации* φ.

Пусть, напротив, семейство $\boldsymbol{S}$ вполне перечислимо, нетривиально и содержит о. Выберем номер b какого-нибудь объекта, не входящего в $\boldsymbol{S}$, и рассмотрим функцию

$$f(x) = \begin{cases} b, & \text{если } x \in \varphi^{-1}\boldsymbol{S}, \\ \text{неопр.}, & \text{если } x \notin \varphi^{-1}\boldsymbol{S}. \end{cases}$$

Согласно п. 6.3 эта функция частично рекурсивная. Поэтому найдется общерекурсивная функция $g(x)$ такая, что

$$\varphi g(x) = \begin{cases} \varphi b, & \text{если } x \in \varphi^{-1}\boldsymbol{S}, \\ \text{о}, & \text{если } x \notin \varphi^{-1}\boldsymbol{S}. \end{cases}$$

Следовательно,

$$x \in \varphi^{-1}\boldsymbol{S}' \Longleftrightarrow g(x) \in \varphi^{-1}\boldsymbol{S}.$$

Это означает, что множество $\varphi^{-1}\boldsymbol{S}'$ $\boldsymbol{m}$-сводимо к множеству $\varphi^{-1}\boldsymbol{S}$. Но последнее множество рекурсивно перечислимо. Значит, множество $\varphi^{-1}\boldsymbol{S}'$ также рекурсивно перечислимо (п. 8.1). Из рекурсивной перечислимости взаимно дополнительных множеств вытекает их рекурсивность. Поэтому семейство $\boldsymbol{S}$ вполне рекурсивно, что противоречит теореме 1.

Теорема 3. *Каждое вполне перечислимое нетривиальное семейство* $\boldsymbol{S}$ *объектов совокупности* $\boldsymbol{U}$, *имеющей полную нумерацию* φ, *является вполне креативным.*

Пусть $a \in \varphi^{-1}\boldsymbol{S}$. Обозначим через $\boldsymbol{T}$ какое-нибудь креативное множество, например, из числа тех, которые были построены в п. 8.2. Рассмотрим вспомогательную функцию

$$f(x)=\begin{cases} a, & \text{если } x\in \boldsymbol{T},\\ \text{неопр.}, & \text{если } x\notin \boldsymbol{T}.\end{cases}$$

Эта функция частично рекурсивная и потому существует общерекурсивная функция $g(x)$, для которой

$$\varphi g(x)=\begin{cases} \varphi a, & \text{если } x\in \boldsymbol{T},\\ \text{o}, & \text{если } x\notin \boldsymbol{T}.\end{cases}$$

Согласно предыдущей теореме $\text{o}\notin\boldsymbol{S}$. Поэтому

$$x\in\boldsymbol{T} \Leftrightarrow g(x)\in\varphi^{-1}\boldsymbol{S}.$$

Мы видим, следовательно, что креативное множество $\boldsymbol{T}$ $\boldsymbol{m}$-сводится к рекурсивно перечислимому множеству $\varphi^{-1}\boldsymbol{S}$. Поэтому множество $\varphi^{-1}\boldsymbol{S}$ креативно.

Теорема 4. *Пусть в совокупности* $\boldsymbol{U}$, *имеющей полную нумерацию* φ, *семейство* $\{\text{o}\}'$ *всех неособых объектов вполне перечислимо. Тогда каждое семейство* $\boldsymbol{S}$ *объектов* $\boldsymbol{U}$, *содержащее* o *и отличное от* $\boldsymbol{U}$, *является вполне продуктивным.*

Рассмотрим частично рекурсивную функцию

$$f(x)=\begin{cases} b, & \text{если } x\in\varphi^{-1}\{\text{o}\}',\\ \text{неопр.}, & \text{если } x\in\varphi^{-1}\text{o},\end{cases}$$

где $\varphi b\in\boldsymbol{S}$. Из полноты нумерации φ вытекает, что найдется общерекурсивная функция $g(x)$, для которой

$$\varphi g(x)=\begin{cases} \varphi b, & \text{если } x\in\varphi^{-1}\{\text{o}\}',\\ \text{o}, & \text{если } x\in\varphi^{-1}\text{o},\end{cases}$$

и, значит,

$$x \in \varphi^{-1}o \Longleftrightarrow g(x) \in \varphi^{-1}S.$$

Согласно теореме 3 множество $\varphi^{-1}o$ продуктивно. Но это множество m-сводится к $\varphi^{-1}S$. Следовательно, $\varphi^{-1}S$ продуктивно, что и требовалось.

Известно (п. 7.2), что в нумерации Клини семейство $\{o\}'$ вполне перечислимое. Поэтому из теоремы 4, например, следует, что совокупность всех клиниевских номеров нигде не определенной функции является продуктивным множеством. Более общо: продуктивной является совокупность всех клиниевских номеров функций любого семейства, лишь бы это семейство включало нигде не определенную функцию и не было семейством вообще всех частично рекурсивных функций.

Теоремы 1—4 хотя и доставляют небольшую информацию о строении вполне перечислимых семейств, но зато для всех полных нумераций. Строение вполне перечислимых семейств для нумераций Клини и Поста известно гораздо более подробно (см. задачи 9, 10 из § 7).

Дополнения, примеры и задачи

1. Пусть $B(x, y)$ — частично рекурсивная функция. Показать (и тем самым закончить формальное доказательство теоремы 1 из п. 9.2), что функция $g(x)$, определенная схемой

$$\begin{aligned} &g(0)=a, \\ &h(n)=\mu_t(B(n, t) \notin \{g(0), \ldots, g(n)\}), \qquad (\alpha) \\ &g(n+1)=B(n, h(n)), \end{aligned}$$

частично рекурсивная.

2. Пусть функция $B(x, y)$ в предыдущей задаче примитивно рекурсивная и удовлетворяет условию разнозначности: из $y \neq z$ вытекает $B(x, y) \neq B(x, z)$. Тогда функция $g(x)$, определенная схемой (α), также примитивно рекурсивная.

3. Провести формальное доказательство рекурсивности функции $h(x)$, построенной в процессе доказательства теоремы 2 из п. 9.2. Показать, что если заданные односводящие функции $f(x)$, $g(x)$ примитивно рекурсивные, то функция $h(x)$, осуществляющая изоморфизм, также примитивно рекурсивная (см. [51]).

4. Пусть частично рекурсивная функция f имеет рекурсивную область определенности. Тогда каждое множество чисел, содержащее все клиниевские номера f и не содержащее ни одного номера какого-нибудь частично рекурсивного расширения f, является продуктивным (Р а й с [81], см. также [52]).

5. Множество чисел заведомо продуктивно, если оно содержит все клиниевские номера некоторой частично рекурсивной функции f и в то же время не содержит ни одного клиниевского номера ни одной конечно определенной функции, график которой содержится в графике f (М а л ь ц е в [52]).

6. Дополнение вполне перечислимой совокупности рекурсивно перечислимых множеств $\mathfrak{S}$ тогда и только тогда вычислимо (определение вычислимости см. п. 7.4), когда $\mathfrak{S}$ есть совокупность всех надмножеств некоторой системы конечных множеств, стандартные номера которых составляют рекурсивное множество. Аналогичное утверждение справедливо и для дополнения вполне перечислимой совокупности частично рекурсивных функций (Р а й с [82]).

7. Показать, что нумерации Клини и Поста не изоморфны (М а л ь ц е в [52]).

8. Семейство объектов $\mathfrak{S}$ нумерованной совокупности, имеющей нумерацию α, называется *внутренне продуктивным*, если существует такая общерекурсивная функция $g(x)$, что для каждого n

$$\pi_n \neq \varnothing \,\&\, \alpha\pi_n \subseteq \mathfrak{S} \Rightarrow \alpha g(n) \in \mathfrak{S} \,\&\, \alpha g(n) \notin \alpha\pi_n.$$

Показать, что никакое внутренне продуктивное семейство в нумерациях Клини и Поста не может быть вычислимым.

9. Показать, что в нумерации Клини каждое семейство частично рекурсивных функций с бесконечной областью определенности, содержащее частичные характеристические функции всех бесконечных рекурсивных множеств, является внутренне продуктивным (М а л ь ц е в [52]).

10. Если семейство частично рекурсивных функций $\mathfrak{S}$ имеет полную вычислимую нумерацию с особым объектом $o = f(x)$, то значения каждой функции из $\mathfrak{S}$ совпадают с значениями $f(x)$ для всех x из области определенности f.

11. Каждая конечная совокупность одноместных частично рекурсивных функций, содержащая нигде не определенную функцию, обладает вычислимой полной нумерацией.

§ 10. УНИВЕРСАЛЬНЫЕ И КРЕАТИВНЫЕ СИСТЕМЫ МНОЖЕСТВ

Понятия m-универсальности и креативности, определенные в § 8 для отдельно взятых числовых множеств, естественно распространяются на пары, тройки, четверки, вообще n-ки множеств. С каждой n-кой множеств $\alpha_1, \ldots, \alpha_n$ однозначно связываем нумерацию семейства символических объектов $\boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$, называя номерами объекта $\boldsymbol{a}_i$ все числа из α_i $(i = 1, \ldots, n)$. Эти нумерации оказываются полными тогда и только тогда, когда n-ки множеств m-универсальны. Общие теоремы § 9, примененные к рассматриваемым частным нумерациям, дают основные свойства

m-универсальных n-ок множеств. В заключительной части рассматриваются понятия продуктивности и отделимости для пар множеств.

10.1. m-универсальные системы множеств. По аналогии с m-сводимостью множеств, определенной в п. 8.1, говорят, что последовательность множеств $\beta_1, \ldots, \beta_n$ m-сводится к последовательности множеств $\alpha_1, \ldots, \alpha_n$, если существует такая рекурсивная функция $f(x)$, что для всех x

$$x \in \beta_i \Leftrightarrow f(x) \in \alpha_i \qquad (i = 1, 2, \ldots, n). \tag{1}$$

Как и ранее, легко убеждаемся, что отношение m-сводимости n-ок множеств транзитивно.

Система множеств $\langle \alpha_1, \ldots, \alpha_n \rangle$ называется *m-универсальной n-кой* множеств, если выполнены следующие условия:

а) множества $\alpha_1, \ldots, \alpha_n$ попарно не имеют общих элементов;

б) каждое из множеств $\alpha_1, \ldots, \alpha_n$ рекурсивно перечислимо;

в) каждая n-ка $\langle \beta_1, \ldots, \beta_n \rangle$ попарно не пересекающихся рекурсивно перечислимых множеств m-сводима к n-ке $\langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle$.

Для $n = 1$ это определение обращается в определение m-универсального множества (п. 8.1). С другой стороны, в определении m-универсальных n-ок множеств не требуется, чтобы множества $\beta_1, \ldots, \beta_n$ были непустыми. Поэтому из m-универсальности какой-нибудь системы множеств $\alpha_1, \ldots$ $\ldots, \alpha_n$ вытекает m-универсальность каждой ее подсистемы $\alpha_{i_1}, \ldots, \alpha_{i_2}, \ldots, \alpha_{i_s}$ и, в частности, m-универсальность каждого отдельного множества α_i, входящего в m-универсальную систему.

Обратное, конечно, неверно: далеко не каждая система, состоящая из n попарно не пересекающихся m-универсальных множеств, является m-универсальной n-кой множеств (см. п. 10.3).

Заметим еще, что объединение всех множеств m-универсальной n-ки не может совпадать с натуральным рядом. Действительно, противное означало бы, что m-универсальное множество α_1 имеет рекурсивно перечислимое дополнение $\alpha_2 \cup \ldots \cup \alpha_n$, чего быть не может, так как дополнение m-универсального множества не рекурсивно.

Из определения $\mathfrak{m}$-универсальных систем множеств непосредственно вытекают следующие их свойства. Пусть $\langle\alpha_1, \ldots, \alpha_n\rangle$ — $\mathfrak{m}$-универсальная система. Тогда

а) *для любой частично рекурсивной функции* $f(x)$ *множества* $f^{-1}(\alpha_1), \ldots, f^{-1}(\alpha_n)$ *попарно не пересекаются и рекурсивно перечислимы;*

б) *для каждой системы* $\beta_1, \ldots, \beta_n$ *попарно не пересекающихся рекурсивно перечислимых множеств существует рекурсивная функция* $f(x)$, *для которой* $\beta_1 = f^{-1}(\alpha_1), \ldots$ $\ldots, \beta_n = f^{-1}(\alpha_n)$.

Обратное также верно: если система $\alpha_1, \ldots, \alpha_n$ каких-то множеств обладает свойствами а), б), то она $\mathfrak{m}$-универсальна.

Свойство б) и послужило первоначальным поводом к введению названия «*$\mathfrak{m}$-универсальные*» системы.

Пусть n — заданное положительное натуральное число. Обозначим через A совокупность каких-то вспомогательных фиксированных объектов $a_0, a_1, \ldots, a_n$. С каждой n-кой $\alpha_1, \ldots, \alpha_n$ попарно не пересекающихся и в сумме не исчерпывающих натурального ряда множеств мы свяжем теперь следующую нумерацию φ совокупности A:

$$\varphi x = \begin{cases} a_i, & \text{если } x \in \alpha_i \quad (i = 1, \ldots, n), \\ a_0, & \text{если } x \notin \alpha_1 \cup \ldots \cup \alpha_n. \end{cases} \tag{2}$$

Нумерацию φ и систему множеств $\alpha_1, \ldots, \alpha_n$ мы далее будем называть *ассоциированными*.

Сопоставляя определения сводимости нумераций (п. 9.2) и $\mathfrak{m}$-сводимости n-ок множеств, видим, что *нумерация* φ *совокупности* A *тогда и только тогда сводима к нумерации* ψ *той же совокупности, когда система множеств, ассоциированная с первой нумерацией,* $\mathfrak{m}$-*сводима к системе множеств, ассоциированной с нумерацией* ψ.

Системы множеств $\langle\alpha_1, \ldots, \alpha_n\rangle$ и $\langle\beta_1, \ldots, \beta_n\rangle$ называются *$\mathfrak{m}$-эквивалентными*, если каждая из них $\mathfrak{m}$-сводима к другой. Из только что сделанного замечания о связи между $\mathfrak{m}$-сводимостью систем и сводимостью ассоциированных нумераций вытекает, что нумерации φ, ψ совокупности A тогда и только тогда эквивалентны, когда $\mathfrak{m}$-эквивалентны ассоциированные с ними системы множеств.

Нижеследующая теорема делает явным тесное родство между понятиями полноты нумерации и $\mathfrak{m}$-универсальности системы множеств.

Теорема 1. *Если φ — полная нумерация совокупности $\boldsymbol{A}$, состоящей из $n+1$ объекта, и множество $\varphi^{-1}\boldsymbol{a}_i$ номеров каждого неособого объекта $\boldsymbol{a}_i$ ($i=1, \ldots, n$) из $\boldsymbol{A}$ рекурсивно перечислимо, то ассоциированная система множеств $\varphi^{-1}\boldsymbol{a}_1, \ldots, \varphi^{-1}\boldsymbol{a}_n$ $\boldsymbol{m}$-универсальна.*

Обратно, если система множеств $\alpha_1, \ldots, \alpha_n$ $\boldsymbol{m}$-универсальна, то ассоциированная нумерация φ полная и совокупность номеров каждого неособого элемента этой нумерации является рекурсивно перечислимой.

Докажем первое утверждение. По условию, множества $\varphi^{-1}\boldsymbol{a}_1, \ldots, \varphi^{-1}\boldsymbol{a}_n$ рекурсивно перечислимые и попарно непересекающиеся. Пусть $\beta_1, \ldots, \beta_n$ — какая-либо другая система непересекающихся рекурсивно перечислимых множеств. Вводим вспомогательную частичную функцию

$$F(x)=\begin{cases} c_i, & \text{если} \quad x \in \beta_i \quad (i=1, \ldots, n), \\ \text{не определена для остальных } x, \end{cases}$$

где $c_1, \ldots, c_n$ — какие-то фиксированные числа из множеств $\varphi^{-1}\boldsymbol{a}_1, \ldots, \varphi^{-1}\boldsymbol{a}_n$. Согласно п. 6.3 функция $F(x)$ частично рекурсивна. Поскольку нумерация φ полная, то найдется общерекурсивная функция $f(x)$, такая, что

$$\varphi f(x)=\begin{cases} \varphi c_i=\boldsymbol{a}_i, & \text{если} \quad x \in \beta_i, \\ \mathrm{o}=\boldsymbol{a}_0, & \text{если} \quad x \notin \beta_1 \cup \ldots \cup \beta_n, \end{cases}$$

то есть

$$x \in \beta_i \Longleftrightarrow f(x) \in \varphi^{-1}\boldsymbol{a}_i \quad (i=1, \ldots, n).$$

Значит, система $\beta_1, \ldots, \beta_n$ $\boldsymbol{m}$-сводится к системе $\{\varphi^{-1}\boldsymbol{a}_i\}$ и система $\varphi^{-1}\boldsymbol{a}_1, \ldots, \varphi^{-1}\boldsymbol{a}_n$ $\boldsymbol{m}$-универсальна.

Докажем обратное. Пусть $F(x)$ — произвольная частично рекурсивная функция. По условию, система множеств $F^{-1}(\alpha_1), \ldots, F^{-1}(\alpha_n)$ $\boldsymbol{m}$-сводится некоторой рекурсивной функцией $f(x)$ к системе $\alpha_1, \ldots, \alpha_n$, то есть

$$F(x) \in \alpha_i \Longleftrightarrow f(x) \in \alpha_i \qquad (i=1, \ldots, n).$$

Сравнивая это с формулами (2), определяющими ассоциированную нумерацию, видим, что

$$\varphi f(x)=\begin{cases} \varphi F(x), & \text{если} \quad x \in \alpha_i \quad (i=1, \ldots, n), \\ \boldsymbol{a}_0, & \text{если} \quad x \notin \alpha_1 \cup \ldots \cup \alpha_n, \end{cases}$$

и, следовательно, нумерация φ полная.

Теорема 2. *Если в какой-нибудь совокупности* $\boldsymbol{U}$ *с полной нумерацией* φ *имеется конечная система* $\boldsymbol{U}_1, \ldots, \boldsymbol{U}_n$ *попарно не пересекающихся вполне перечислимых семейств объектов, то система множеств* $\varphi^{-1}\boldsymbol{U}_1, \ldots, \varphi^{-1}\boldsymbol{U}_n$ *является* $\boldsymbol{m}$*-универсальной.*

В самом деле, обозначим через λ разбиение совокупности $\boldsymbol{U}$ на классы $\boldsymbol{U}_1, \ldots, \boldsymbol{U}_n$ и класс $\boldsymbol{U}_0 = \boldsymbol{U} - (\boldsymbol{U}_1 \bigcup \ldots \bigcup \boldsymbol{U}_n)$ (см. п. 9.1). Фактор-нумерация $\varphi_0 = \varphi/\lambda$ является полной нумерацией конечной совокупности $\boldsymbol{A} = \{\boldsymbol{U}_0, \boldsymbol{U}_1, \ldots, \boldsymbol{U}_n\}$, причем совокупности номеров всех неособых элементов здесь рекурсивно перечислимы (так как они совпадают с множествами $\varphi^{-1}\boldsymbol{U}_i$, $i = 1, \ldots, n$). Согласно теореме 1 отсюда вытекает, что система множеств $\varphi^{-1}\boldsymbol{U}_1, \ldots, \varphi^{-1}\boldsymbol{U}_n$ $\boldsymbol{m}$-универсальная.

Теорема 2 дает возможность весьма просто находить конкретные $\boldsymbol{m}$-универсальные системы множеств. Рассмотрим, например, нумерацию Клини $\varkappa$. Пусть $\boldsymbol{U}_i$ — семейство всех одноместных частично рекурсивных функций $f(x)$, которые удовлетворяют условию $f(0) = i$. Семейства $\boldsymbol{U}_i$ вполне перечислимые и попарно не пересекаются. Обозначая через α_i совокупность всех клиниевских номеров функций из $\boldsymbol{U}_i$, видим, что согласно теореме 2 система множеств $\alpha_1, \ldots \alpha_n$ является $\boldsymbol{m}$-универсальной для каждого значения $n = 1, 2, \ldots$

Системы множеств $\alpha_1, \ldots, \alpha_n$ и $\beta_1, \ldots, \beta_n$ называются *рекурсивно изоморфными*, если одна из них переводится в другую подходящим рекурсивным взаимно однозначным отображением натурального ряда на себя, то есть если одна из систем $\boldsymbol{m}$-сводится к другой при помощи подходящей рекурсивной функции, осуществляющей взаимно однозначное отображение натурального ряда на себя.

Теорема 3. *Для каждого* n *все* $\boldsymbol{m}$*-универсальные* n*-ки множеств изоморфны друг другу* (А. Мучник [64], Шмульян [121]).

Действительно, из определения $\boldsymbol{m}$-универсальности непосредственно следует, что все $\boldsymbol{m}$-универсальные n-ки множеств $\boldsymbol{m}$-эквивалентны друг другу. Из $\boldsymbol{m}$-эквивалентности систем множеств следует эквивалентность ассоциированных нумераций. Так как эти нумерации в рассматриваемом случае полные (теорема 1), то они будут

и изоморфными (теорема 5, п. 9.3). Но тогда изоморфными будут и соответствующие n-ки множеств.

10.2. Креативные системы множеств. Понятие креативности, определенное в п. 8.2 для отдельно взятых числовых множеств, распространяется на пары, тройки и т. д. множеств следующим образом.

Система n числовых множеств $\alpha_1, \ldots, \alpha_n$ называется *креативной n-кой множеств*, если эти множества попарно не пересекаются, рекурсивно перечислимы и если существует такая рекурсивная функция $p(x_1, \ldots, x_n)$, что для каждой системы $\boldsymbol{\pi}_{x_1}, \ldots, \boldsymbol{\pi}_{x_n}$ попарно не пересекающихся рекурсивно перечислимых множеств, удовлетворяющих условиям дополнительности $\alpha_1 \cap \boldsymbol{\pi}_{x_1} = \varnothing, \ldots, \alpha_n \cap \boldsymbol{\pi}_{x_n} = \varnothing$, имеет место соотношение

$$p(x_1, \ldots, x_n) \notin (\alpha_1 \cup \boldsymbol{\pi}_{x_1}) \cup \ldots \cup (\alpha_n \cup \boldsymbol{\pi}_{x_n}). \qquad (1)$$

При $n = 1$ это определение обращается в определение креативности множества α_1, введенное в п. 8.2. Теорема о совпадении классов креативных и m-универсальных множеств также переносится на m-универсальные и креативные n-ки множеств. Однако, чтобы получить несколько более сильные результаты, мы разобьем теорему о совпадении классов креативных и универсальных n-ок на две части.

Теорема 1. *Для каждой универсальной n-ки множеств $\alpha_1, \ldots, \alpha_n$ существует такая рекурсивная функция $g(x_1, \ldots, x_n)$, что*

$$g(x_1, \ldots, x_n) \in \bigcup_{i=1}^{n} (\boldsymbol{\pi}_{x_i} \cap \alpha_i) \cup \bigcap_{i=1}^{n} (\boldsymbol{\pi}'_{x_i} \cap \alpha'_i) \qquad (2)$$

для всех $x_1, \ldots, x_n$.

Если $\boldsymbol{\pi}_{x_i} \cap \alpha_i = \varnothing$ $(i = 1, \ldots, n)$, то условие (2) переходит в условие креативности (1). Поэтому из теоремы 1 следует, что *все универсальные n-ки множеств являются креативными.*

Для доказательства теоремы 1 выберем в каждом множестве α_i какое-нибудь число c_i $(i = 1, \ldots, n)$. Рассмотрим совместное продолжение $f(x_1, \ldots, x_n, x)$ постоянных функций $f_i(x) = c_i$ $(i = 1, \ldots, n)$, определенное в теореме о совместном продолжении (п. 7.3) и обладающее следующими свойствами:

а) функция $f(x_1, \ldots, x_n, x)$ частично рекурсивна по совокупности всех своих аргументов;

б) для каждых фиксированных $x_1, \ldots, x_n$ область определенности одноместной функции $f(x_1, \ldots, x_n, x)$ есть $\boldsymbol{\pi}_{x_1} \bigcup \ldots \bigcup \boldsymbol{\pi}_{x_n}$;

в) $f(x_1, \ldots, x_n, x)$ может принимать лишь значения $c_1, \ldots, c_n$ и

$$f(x_1, \ldots, x_n, x) = c_i \Rightarrow x \in \boldsymbol{\pi}_{x_i} \qquad (i = 1, \ldots, n).$$

Рассмотрим теперь нумерацию φ вспомогательной совокупности $A = \{\boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}$, ассоциированную с системой $\alpha_1, \ldots, \alpha_n$. Так как система $\alpha_1, \ldots, \alpha_n$ универсальная, то нумерация φ полная, и, следовательно, найдется рекурсивная функция $g(x_1, x_2, \ldots, x_n)$ (п. 9.3, теорема 4), удовлетворяющая условию

$$\varphi g(x_1, \ldots, x_n) = \varphi f(x_1, \ldots, x_n, g(x_1, \ldots, x_n)), \tag{3}$$

если $f(x_1, \ldots, x_n, g)$ определено, то есть если

$$g(x_1, \ldots, x_n) \in \boldsymbol{\pi}_{x_1} \bigcup \ldots \bigcup \boldsymbol{\pi}_{x_n},$$

и условию

$$\varphi g(x_1, \ldots, x_n) = \boldsymbol{a}_0, \tag{4}$$

если $g(x_1, \ldots, x_n) \notin \boldsymbol{\pi}_{x_1} \bigcup \ldots \bigcup \boldsymbol{\pi}_{x_n}$.

В случае (3) из свойства в) следует, что при подходящем j, $1 \leqslant j \leqslant n$, имеем $f(x_1, \ldots, x_n, g) = c_j$ и $g(x_1, \ldots, x_n) \in \boldsymbol{\pi}_{x_j}$. Равенство (3) дает $\varphi g = \varphi c_j$, откуда $g \in \alpha_j$ и потому $g \in \boldsymbol{\pi}_{x_j} \bigcap \alpha_j$.

В случае (4) $\varphi g = \boldsymbol{a}_0$, то есть $g \notin \alpha_1 \bigcup \ldots \bigcup \alpha_n$ и потому $g \notin \bigcup (\boldsymbol{\pi}_{x_i} \bigcup \alpha_i)$, $g \in \bigcap (\boldsymbol{\pi}'_{x_i} \bigcap \alpha'_i)$.

Итак, в случаях (3) и (4) имеет место включение (2), то есть включение (2) имеет место при любых $x_1, \ldots, x_n$, что и требовалось.

Систему рекурсивно перечислимых попарно не пересекающихся множеств $\alpha_1, \ldots, \alpha_n$ условимся называть *слабо креативной*, если существует рекурсивная функция $g(x_1, \ldots, x_n)$, обладающая следующим свойством: если для некоторых $x_1, \ldots, x_n$ все множества $\boldsymbol{\pi}_{x_1}, \ldots, \boldsymbol{\pi}_{x_n}$ пусты за исключением, быть может, одного $\boldsymbol{\pi}_{x_i}$ и это $\boldsymbol{\pi}_{x_i}$ состоит

лишь из одного числа, не входящего в соответствующее α_i, то

$$g(x_1, \ldots, x_n) \notin \bigcup (\pi_{x_i} \bigcup \alpha_i). \tag{5}$$

Отсюда видно, что условие слабой креативности (5) совпадает с условием креативности (1), но при более жестких ограничительных требованиях, налагаемых на числа $x_1, \ldots, x_n$. Поэтому креативные системы заведомо слабо креативны. Из теоремы 1 и нижеследующей теоремы 2 будет видно, что справедливо и обратное.

Теорема 2. *Каждая слабо креативная n-ка множеств $\alpha_1, \ldots, \alpha_n$ является $\boldsymbol{m}$-универсальной.*

Пусть $g(x_1, \ldots, x_n)$ — рекурсивная функция, удовлетворяющая для системы $\alpha_1, \ldots, \alpha_n$ условию (5). Нам надо показать, что любая система $\beta_1, \ldots, \beta_n$ рекурсивно перечислимых попарно не пересекающихся множеств $\boldsymbol{m}$-сводится подходящей рекурсивной функцией $h(x)$ к системе $\alpha_1, \ldots, \alpha_n$. Функция $h(x)$ строится так.

Сначала вместо n-местной продуктивной функции вводим одноместную следующим образом. Согласно п. 7.3 для каждого i существует одноместная рекурсивная функция $g_i(x)$ такая, что

$$\boldsymbol{K}(x, t) = i \iff t \in \pi_{g_i(x)}. \tag{6}$$

Полагаем по определению

$$p(x) = g(g_1(x), \ldots, g_n(x)).$$

Из свойства (5) функции g и формул (6) вытекает, что $p(x)$ удовлетворяет следующему условию. Пусть для некоторого x

$$\boldsymbol{K}(x, t) = i \iff t \in \gamma_i \qquad (i = 1, \ldots, n), \tag{7}$$

причем либо все множества $\gamma_1, \ldots, \gamma_n$ пусты, либо все пусты, кроме некоторого множества γ_i, состоящего лишь из одного числа, и $\alpha_i \bigcap \gamma_i = \varnothing$. Тогда

$$p(x) \notin \bigcup_{j=1}^{n} (\alpha_j \bigcup \gamma_j). \tag{8}$$

Согласно теореме о совместном продолжении (п. 7.3) существует частично рекурсивная функция $\boldsymbol{K}(a, y, z, t)$

от переменных y, z, t, удовлетворяющая условиям

$$\boldsymbol{K}(a, y, z, t) = \begin{cases} i \text{ при } y \in \beta_i, \ t \in \boldsymbol{\pi}_z \quad (i = 1, \ldots, n), \\ \text{не определена для остальных } y, z, t. \end{cases} \tag{9}$$

По теореме Майхила о неподвижной точке найдется рекурсивная функция $f(y)$, для которой

$$p([a, y, f(y)]) = x \Longleftrightarrow x \in \boldsymbol{\pi}_{f(y)}.$$

Это означает просто, что для каждого y множество $\boldsymbol{\pi}_{f(y)}$ состоит лишь из одного числа $p([a, y, f(y)])$.

Покажем теперь, что функция $h(y) = p([a, y, f(y)])$ $\boldsymbol{m}$-сводит систему $\beta_1, \ldots, \beta_n$ к $\alpha_1, \ldots, \alpha_n$. Прежде всего,

$$y \notin \beta_1 \bigcup \ldots \bigcup \beta_n \Rightarrow h(y) \notin \alpha_1 \bigcup \ldots \bigcup \alpha_n. \tag{10}$$

Действительно, согласно (9) из $y \notin \beta_1 \bigcup \ldots \bigcup \beta_n$ вытекает, что $\boldsymbol{K}([a, y, f(y)], t)$ не определено ни для каких t. Поэтому множества $\gamma_1, \ldots, \gamma_n$, определенные по формулам (7), пусты и согласно (8) $p([a, y, f(y)]) \notin \alpha_1 \bigcup \ldots \bigcup \alpha_n$.

С другой стороны, из (9) получаем, что при $y \in \beta_i$ функция $\boldsymbol{K}([a, y, f(y)], t)$ имеет определенное значение лишь для $t = p([a, y, f(y)])$ и значение это равно i. Следовательно, все множества $\gamma_1, \ldots, \gamma_n$, найденные в этом случае по формулам (7), пусты, кроме множества γ_i, состоящего из числа $p([a, y, f(y)])$. Если бы это число не входило в α_i, то согласно (8) мы имели бы соотношение

$$p([a, y, f(y)]) \notin \bigcup_{j=1}^{n} (\alpha_j \bigcup \gamma_j),$$

что невозможно, так как $p([a, y, f(y)]) \in \gamma_i$. Следовательно,

$$y \in \beta_i \Rightarrow h(y) \in \alpha_i \qquad (i = 1, \ldots, n)$$

и, значит, функция $h(y)$ действительно $\boldsymbol{m}$-сводит систему $\beta_1, \ldots, \beta_n$ к системе $\alpha_1, \ldots, \alpha_n$.

10.3. Рекурсивно неотделимые множества. Говорят, что числовое множество γ *отделяет* множество α от множества β, если $\alpha \subseteq \gamma$ и $\gamma \bigcap \beta = \varnothing$. Множество α называется *рекурсивно отделимым* от множества β, если существует рекурсивное множество γ, отделяющее α от β.

Если рекурсивное множество γ отделяет множество α от множества β, то дополнение γ' множества γ рекурсивно

отделяет β от α. Поэтому отношение рекурсивной отделимости симметрично и вместо того, чтобы говорить, что множество α рекурсивно отделимо от множества β, часто говорят, что множества α и β рекурсивно отделимы.

Множества α и β называются *рекурсивно неотделимыми*, если они непересекающиеся и ни одно из них не отделимо рекурсивно от другого.

Каждое рекурсивное множество отделяет рекурсивно самого себя от любого другого не пересекающегося с ним множества. Поэтому рекурсивно неотделимыми могут быть лишь нерекурсивные множества.

Теорема 1. *Рекурсивно перечислимые рекурсивно неотделимые множества существуют.*

Рассмотрим частично рекурсивную функцию $E(x)$, построенную в п. 6.3, которая принимает лишь значения 0,1 и не может быть доопределена до общерекурсивной функции. Обозначим через α совокупность решений уравнения $E(x) = 0$ и через β совокупность решений уравнения $E(x) = 1$. Оба эти множества рекурсивно перечислимые и непересекающиеся. Докажем, что α не отделимо рекурсивно от β. Пусть, напротив, какое-то рекурсивное множество γ отделяет α от β. Тогда характеристическая функция множества γ была бы рекурсивным доопределением функции E в противоречие с установленной ранее рекурсивной недоопределимостью этой функции.

Из определения рекурсивной неотделимости непосредственно вытекают следующие свойства рекурсивно неотделимых множеств:

а) *никакое нерекурсивное множество не может быть рекурсивно отделимым от своего дополнения*;

б) *если множества α, β рекурсивно неотделимы и $\alpha \subseteq \alpha_1$, $\beta \subseteq \beta_1$, $\alpha_1 \cap \beta_1 = \varnothing$, то множества α_1, β_1 также рекурсивно неотделимы.*

Более тонкое свойство рекурсивной отделимости указывает

Теорема 2. *Если непересекающиеся множества α, β имеют рекурсивно перечислимые дополнения, то α и β рекурсивно отделимы.*

Согласно теореме о совместном продолжении (п. 7.3) существует частично рекурсивная функция $f(x)$, определенная на объединении множеств α' и β', принимающая лишь

значения 1, 2 и удовлетворяющая требованиям

$$f(x)=1 \Rightarrow x \in \alpha', \qquad f(x)=2 \Rightarrow x \in \beta'. \qquad (1)$$

По условию $\alpha \cap \beta = \varnothing$ и, значит, $\alpha' \cup \beta'$ совпадает со всем натуральным рядом. Поэтому функция $f(x)$ всюду определенная. Обозначим через α_1 и β_1 множества тех x, для которых $f(x) = 1$ и соответственно $f(x) = 2$. Из рекурсивности функции $f(x)$ следует, что α_1 и β_1 рекурсивны. Из (1) вытекает, что $\alpha_1 \supseteq \alpha'$, $\beta_1 \supseteq \beta'$, то есть $\alpha \subseteq \alpha_1'$, $\beta \subseteq \beta_1'$. Значит, α и β отделяются рекурсивными непересекающимися множествами α_1', β_1'.

Теорема 3. *Никакая m-универсальная пара множеств α, β не может быть отделимой.*

В самом деле, согласно п. 10.2 каждая универсальная пара креативна и потому для множеств α, β существует рекурсивная «продуктивная» функция $g(x, y)$, значения которой подчинены условию: если $\pi_x \cap \pi_y = \varnothing$ и $\alpha \cap \pi_x = \beta \cap \pi_y = \varnothing$, то

$$g(x, y) \notin \alpha \cup \beta \cup \pi_x \cup \pi_y. \qquad (2)$$

Если бы множества α, β были отделимы, то для подходящих x, y множества π_x, π_y были бы взаимно дополнительны и удовлетворяли бы условиям $\alpha \subseteq \pi_x$, $\beta \subseteq \pi_y$, то есть $\alpha \cap \pi_y = \beta \cap \pi_x = \varnothing$. Но тогда соотношение (2) было бы невозможно, так как функция $g(x, y)$ должна иметь определенное значение, а справа стоит множество, совпадающее со всем натуральным рядом.

Наряду с понятием неотделимости множества часто бывает полезным и понятие эффективной неотделимости, определяемое следующим образом.

Непересекающиеся множества α, β называются эффективно неотделимыми, если существует рекурсивная функция $q(x, y)$ такая, что

$$\pi_x \cap \pi_y = \varnothing \;\&\; \pi_x \supseteq \alpha \;\&\; \pi_y \supseteq \beta \Rightarrow q(x, y) \notin \pi_x \cup \pi_y. \qquad (3)$$

Повторяя приведенные выше рассуждения, легко убеждаемся, что каждая эффективно неотделимая пара множеств неотделима в обычном смысле. Более того, если в определении креативной пары множеств опустить требование, чтобы множества пары были рекурсивно перечислимыми,

то условие (3) будет следствием условия (1) из п. 10.2, посредством которого определялось понятие креативности. Таким образом, каждая креативная пара множеств эффективно неотделима. Однако верна и более общая

Теорема 4. *Непересекающиеся рекурсивно перечислимые множества тогда и только тогда эффективно неотделимы, когда они образуют креативную пару.*

То, что креативная пара эффективно неотделима, уже было установлено. Поэтому пусть α, β — эффективно неотделимая пара рекурсивно перечислимых множеств. Мы хотим показать, что в этом случае пара α, β креативна.

Так как α и β рекурсивно перечислимые, то существуют числа a, b, для которых при любом x

$$\alpha \bigcup \pi_x = \pi_{[a, x]}, \quad \beta \bigcup \pi_x = \pi_{[b, x]}.$$

Пусть $\pi_x \bigcap \pi_t = \pi_s \bigcap \alpha = \pi_t \bigcap \beta = \varnothing$. Тогда $\pi_{[a, t]} \supseteq \alpha$; $\pi_{[b, s]} \supseteq \beta$ и $\pi_{[a, t]} \bigcap \pi_{[b, s]} = \varnothing$. Следовательно, из (3) имеем

$$q([a, t], [b, s]) \notin \alpha \bigcup \pi_s \bigcup \beta \bigcup \pi_t.$$

Это показывает, что пара множеств α, β креативная, имеющая в качестве продуктивной функции функцию

$$p(s, t) = q([a, t], [b, s]).$$

Итак, согласно теореме 4 для пар рекурсивно перечислимых множеств понятия креативности и эффективной неотделимости равносильны. Мы сейчас воспользуемся этим обстоятельством, чтобы доказать

Следствие 1. *Пусть множества* α, β, *образующие креативную пару, отделены рекурсивно перечислимыми множествами* α_1, β_1, *то есть* $\alpha \subseteq \alpha_1$, $\beta \subseteq \beta_1$, $\alpha_1 \bigcap \beta_1 = \varnothing$. *Тогда множества* α_1, β_1 *также образуют креативную пару.*

В самом деле, если функция $q(x, y)$ удовлетворяет для множеств α, β требованиям (3), то эта же функция будет удовлетворять требованиям (3) для множеств α_1, β_1. Следовательно, множества α_1, β_1 эффективно неотделимы, а значит, они образуют креативную пару.

Дополнения, примеры и задачи

1. Теория m-универсальных n-ок множеств без труда переносится и на бесконечные системы множеств ([39], [52]).

Вводятся следующие определения: последовательность множеств α_0, α_1, . . . называется m-сводящейся к последовательности

$\beta_0, \beta_1, \ldots$, если для подходящей рекурсивной функции $f(x)$

$$x \in \alpha_i \Longleftrightarrow f(x) \in \beta_i \qquad (i=0, 1, \ldots).$$

Последовательности называются m-эквивалентными, если каждая из них m-сводится к другой. Если $h(x)$ — рекурсивная функция, то последовательность множеств $\langle \pi_{h(0)}, \pi_{h(1)}, \ldots \rangle$ называется р. п. последовательностью. Р. п. последовательность, состоящая из попарно не пересекающихся множеств, называется *серией*. Серия называется m-универсальной, если к ней m-сводится любая серия. С каждой серией $\alpha_1, \alpha_2, \ldots$, состоящей из непустых множеств, не исчерпывающих в совокупности натурального ряда, ассоциируем нумерацию φ совокупности натуральных чисел, полагая по определению

$$\varphi n = i \Longleftrightarrow n \in \alpha_i \qquad (n, i = 0, 1, \ldots),$$

где $\alpha_0 = N - (\alpha_1 \cup \alpha_2 \cup \ldots)$. Изложенными в данном параграфе методами легко доказываются следующие утверждения:

а) серия тогда и только тогда m-универсальна, когда ассоциированная нумерация полная; б) все m-универсальные серии изоморфны друг другу; в) если в полной нумерации φ для некоторой рекурсивной функции $h(x)$ множества $\varphi^{-1} h(x)$ непусты и составляют серию, то эта серия m-универсальна.

2. Пусть U — совокупность с нумерацией φ и выделенным элементом о, называемым *особым*. *Ведущей* функцией нумерации φ называется частичная одноместная функция $f(x)$, определенная на множестве $\varphi^{-1}(U - о)$ и удовлетворяющая следующему требованию: $f(x) = f(y) \Longleftrightarrow \varphi x = \varphi y$ для всех x, y из области определенности f. Нумерация φ тогда и только тогда ассоциирована с некоторой серией, когда для φ существует частично рекурсивная ведущая функция. В терминах ведущих функций можно сформулировать понятие креативности серии и доказать соответствующие теоремы (см. Клив [39]).

3. Пусть существует частично рекурсивная функция $f(x, y)$ такая, что из $\pi_x \supseteq \alpha$ & $\pi_y \supseteq \beta$ & $\pi_x \cap \pi_y = \varnothing$ следует, что $f(x, y)$ определена и $f(x, y) \notin \pi_x \cup \pi_y$ (α, β фиксированы). Тогда α, β эффективно неотделимы (А. А. Мучник [63]).

4. Для любых x, y множества $\pi_x - \pi_y$ и $\pi_y - \pi_x$ отделимы рекурсивно перечислимыми множествами.

5. Рекурсивно перечислимые множества α, β называются (А. А. Мучник [63]) *сильно неотделимыми*, если выполнены следующие условия: а) $\alpha \cap \beta = \varnothing$; б) дополнение множества $\alpha \cup \beta$ бесконечное; в) для любого рекурсивно перечислимого π_x из $\alpha \cap \pi_x = \varnothing$ следует, что $\pi_x - \beta$ конечно (или пусто) и из $\beta \cap \pi_x = \varnothing$ следует, что $\pi_x - \alpha$ конечно. Доказать, что сильно неотделимые множества не отделимы рекурсивными множествами, но не могут быть эффективно неотделимыми. Построить пример сильно неотделимых множеств.

6. Понятия рекурсивной неотделимости и эффективной неотделимости можно распространить и на пары пересекающихся множеств (А. А. Мучник [63], Р. Шмульян [121]). Множества

α, β (возможно, пересекающиеся) называются рекурсивно неотделимыми, если не существует рекурсивных множеств α_1, β_1, удовлетворяющих требованиям: $\alpha_1 \supseteq \alpha$, $\beta_1 \supseteq \beta$, $\alpha_1 \cap \beta_1 = \alpha \cap \beta$. Аналогично, множества α, β называются эффективно неотделимыми, если существует рекурсивная функция $f(x, y)$ такая, что для всех x, y

$$\boldsymbol{\pi}_x \supseteq \alpha \,\&\, \boldsymbol{\pi}_y \supseteq \beta \,\&\, \boldsymbol{\pi}_x \cap \boldsymbol{\pi}_y = \alpha \cap \beta \Rightarrow f(x, y) \notin \boldsymbol{\pi}_x \cup \boldsymbol{\pi}_y.$$

Пара множеств α, β называется продуктивной, если существует рекурсивная функция $p(x, y)$, удовлетворяющая требованиям

$$\boldsymbol{\pi}_x \subseteq \alpha \,\&\, \boldsymbol{\pi}_y \subseteq \beta \,\&\, \boldsymbol{\pi}_x \cap \boldsymbol{\pi}_y = \varnothing \Rightarrow p(x, y) \in (\alpha \cap \beta) - (\boldsymbol{\pi}_x \cup \boldsymbol{\pi}_y).$$

Показать, что из m-сводимости эффективно неотделимой пары α, β к паре γ, δ вытекает эффективная неотделимость пары γ, δ (пары могут пересекаться!). Далее, если α', β' — продуктивная пара, то α, β эффективно неотделимы. Если пара рекурсивно перечислимых множеств α, β эффективно неотделимая, то пара α', β' продуктивная.

ГЛАВА V

АЛГОРИТМЫ И МАШИНЫ ТЬЮРИНГА

В предшествующих главах подробно изучались свойства частично рекурсивных функций. Обосновывая необходимость этого изучения, мы приняли в качестве постулата тезис Чёрча о том, что совокупность частично рекурсивных функций совпадает с совокупностью всех функций, вычислимых посредством алгоритмов. В последующем изложении тезис Чёрча для доказательства теорем не использовался. Напротив, во многих случаях наша основная цель состояла в том, чтобы доказать частичную рекурсивность функций, заданных посредством сложных процессов алгоритмического характера. То, что эта цель всегда и без особого труда достигалась, может служить главным подтверждением тезиса Чёрча.

Как уже говорилось, мы не ставили себе целью дать всеобщее определение интуитивному понятию алгоритма, но нами до сих пор не были описаны вообще никакие классы процессов алгоритмического характера, достаточные для вычисления всех частично рекурсивных функций. Правда, это сделать было нетрудно, так как уже простой анализ понятия частичной рекурсивности приводит к одному из таких классов. Однако этот класс получается довольно сложным. В настоящей и следующей главах будет описан ряд важных и естественных классов алгоритмов и каждый раз будет показано, что совокупность функций, вычислимых посредством алгоритмов рассматриваемого класса, в точности совпадает с совокупностью частично рекурсивных функций. Это доставит нам ряд новых подтверждений тезиса Чёрча и вместе с тем даст много новых тонких свойств частично рекурсивных функций.

§ 11. СЛОВАРНЫЕ МНОЖЕСТВА И ФУНКЦИИ

Описание важнейших классов алгоритмов производится в терминах теории слов в некотором алфавите. Начала ее были изложены в § 1. Теперь мы хотим продвинуть эту теорию несколько дальше с тем, чтобы понятия рекурсивности и частичной рекурсивности, определенные ранее для числовых функций и множеств, распространить на словарные множества и функции.

11.1. Словарные множества. Пусть $A \doteq \{a_1, \ldots \ldots, a_p\}$ — какой-нибудь конечный набор символов. Обозначим через $\mathfrak{S}_A$ совокупность всех слов в алфавите A, включая и пустое слово Λ. Всевозможные подмножества совокупности $\mathfrak{S}_A$, в том числе пустое подмножество $\varnothing$, будут называться *словарными* множествами в алфавите A. Мы хотим определить понятия рекурсивных и рекурсивно перечислимых словарных множеств. Это будет сделано двумя путями: сначала при помощи нумерации слов, а в следующем разделе и более прямым способом.

Из многих возможных нумераций совокупности $\mathfrak{S}_A$ мы остановимся на так называемой *алфавитной* нумерации, определяемой следующим образом.

Номер пустого слова Λ полагаем равным нулю. Далее в каком-либо порядке числами $1, 2, \ldots, p$ нумеруем символы алфавита. Пусть a_i обозначает символ с номером i. По определению номером слова $\mathfrak{a} = a_{i_s} \ldots a_{i_1} a_{i_0}$ называем число

$$c(\mathfrak{a}) = i_0 + i_1 p + \ldots + i_s p^s. \qquad (1)$$

Символами $c(\mathfrak{a})$, αn условимся обозначать алфавитный номер слова $\mathfrak{a}$ и соответственно слово, имеющее номер n. Так как при фиксированном p каждое положительное число представимо и лишь одним способом в виде

$$n = i_0 + i_1 p + \ldots + i_s p^s \quad (1 \leqslant i_\lambda \leqslant p), \qquad (2)$$

то каждое число есть алфавитный номер одного и только одного слова совокупности $\mathfrak{S}_A$. Разложение (2) иногда называется *p-ичным разложением числа n с цифрами* $1, 2, \ldots, p$ в отличие от обычного p-ичного разложе-

ния, в котором в качестве коэффициентов допускаются числа 0, 1, ..., $p-1$.

Множество слов $\mathfrak{M}$ в алфавите $\boldsymbol{A}$ называется *примитивно рекурсивным*, *рекурсивным* или *рекурсивно перечислимым*, если соответственно примитивно рекурсивна, рекурсивна или рекурсивно перечислима совокупность алфавитных номеров всех слов из $\mathfrak{M}$.

Принимая во внимание результаты пп. 4.1 и 4.2, непосредственно видим, что

а) каждое конечное множество слов примитивно рекурсивно;

б) объединение и пересечение конечной системы примитивно рекурсивных, рекурсивных или рекурсивно перечислимых множеств слов суть множества того же типа;

с) если словарное множество $\mathfrak{M}$ и его дополнение $\mathfrak{S}_{\boldsymbol{A}}-\mathfrak{M}$ рекурсивно перечислимы, то они рекурсивны.

Говорят, что в алфавите $\boldsymbol{A}$ задана частичная n-местная словарная функция F, если некоторым n-кам $\mathfrak{x}_1, \ldots, \mathfrak{x}_n$ слов из $\mathfrak{S}_{\boldsymbol{A}}$ поставлены в соответствие однозначно определенные слова $F(\mathfrak{x}_1, \ldots, \mathfrak{x}_n)$ в $\boldsymbol{A}$ (ср. п. 1.2). Числовая n-местная частичная функция f называется функцией, представляющей словарную функцию F в нумерации α, если

$$F(\alpha x_1, \ldots, \alpha x_n) = \alpha f(x_1, \ldots, x_n)$$

для всех натуральных $x_1, \ldots, x_n$ и, следовательно,

$$F(\mathfrak{x}_1, \ldots, \mathfrak{x}_n) = \alpha f(c\mathfrak{x}_1, \ldots, c\mathfrak{x}_n), \tag{3}$$

$$f(x_1, \ldots, x_n) = c(F(\alpha x_1, \ldots, \alpha x_n)). \tag{4}$$

Очевидно, каждая частичная словарная функция обладает единственной представляющей функцией и каждая частичная числовая функция представляет определенную словарную функцию.

Частичная словарная функция F называется *примитивно рекурсивной*, *общерекурсивной* или *частично рекурсивной*, если таковой является числовая функция, представляющая F.

Например, постоянные словарные функции представляются постоянными числовыми функциями. Постоянные числовые функции примитивно рекурсивны. Поэтому постоянные словарные функции примитивно рекурсивны.

Докажем еще примитивную рекурсивность простейших словарных функций $S_i(\mathfrak{x})$, определяемых формулами

$$S_i(\mathfrak{x}) = \mathfrak{x}a_i \qquad (i = 1, \ldots, p).$$

Согласно (4) представляющей функцией для $S_i(\mathfrak{x})$ является функция

$$s_i(x) = c(\alpha x \cdot a_i).$$

Из формулы (1) непосредственно видно, что

$$c(\mathfrak{x}a_i) = pc(\mathfrak{x}) + i \tag{5}$$

и потому

$$s_i(x) = px + i. \tag{6}$$

Таким образом, представляющие функции для словарных функций S_i примитивно рекурсивны, а вместе с ними будут примитивно рекурсивными и сами словарные функции.

Аналогичным путем легко доказать примитивную рекурсивность функции $F(\mathfrak{x}, \mathfrak{y}) = \mathfrak{x}\mathfrak{y}$ и многих других, которые нам повстречаются далее. Однако здесь мы это делать не будем, так как в следующем разделе это получится еще более просто.

Введенные только что определения рекурсивности, частичной рекурсивности словарных множеств и функций зависят от первоначальной нумерации алфавитных символов. Кроме того, возникает вопрос: если множество $\mathfrak{M}$ слов в алфавите A, например, рекурсивно, то будет ли $\mathfrak{M}$ рекурсивно и в произвольном расширении алфавита A?

Ясно, что свойство быть рекурсивным множеством слов не зависит ни от нумерации символов алфавита, ни от того, в каком алфавите это множество рассматривается. Оба эти утверждения доказываются аналогично. Для полноты мы изложим доказательство второго из них.

Теорема 1. *Рассмотрим какой-нибудь алфавит $A = \{a_1, \ldots, a_p\}$ и его расширение $A_1 = \{a_1, \ldots, a_p, a_{p+1}, \ldots, a_q\}$. Совокупность $\mathfrak{M}$ некоторых слов в алфавите A тогда и только тогда примитивно рекурсивна, рекурсивна или рекурсивно перечислима в A, когда она имеет тот же тип в A_1.*

Частичная словарная функция $F(\mathfrak{x}_1, \ldots, \mathfrak{x}_n)$, определенная в алфавите A, тогда и только тогда частично рекурсивна в A, когда она частично рекурсивна в A_1.

Всюду определенная на $\mathfrak{S}_A$ словарная функция F тогда и только тогда примитивно рекурсивна или рекурсивна в $\boldsymbol{A}$, когда F может быть доопределена до примитивно рекурсивной или соответственно рекурсивной функции в $\boldsymbol{A}_1$.

Докажем сначала первое утверждение. Обозначим через α, α_1 словарные нумерации совокупностей $\mathfrak{S}_A$, $\mathfrak{S}_{A_1}$. Аналогично, если $\mathfrak{a}$ — слово в алфавите $\boldsymbol{A}$, то через $c(\mathfrak{a})$ обозначим его α-номер, а через $c_1(\mathfrak{a})$—α_1-номер. Покажем, что функция $f(x) = c_1(\alpha x)$ примитивно рекурсивна.

Пусть $x > 0$, $\alpha x = \mathfrak{a}$. Представляем $\mathfrak{a}$ в виде $\mathfrak{b}a_i$. Из (5) и (6) получаем

$$x = pc(\mathfrak{b}) + i,\ c_1(\mathfrak{b}a_i) = qc_1(\mathfrak{b}) + i,$$

откуда

$$f(x) = qf([x/p] \dot{-} \overline{\mathrm{sg}}\,\mathrm{rest}(x, p)) + (x \dot{-} p[x/p]) + p\,\overline{\mathrm{sg}}\,\mathrm{rest}(x, p).$$

Это соотношение можно рассматривать как возвратную рекурсию для $f(x)$ и потому в силу п. 3.2 функция $f(x)$ примитивно рекурсивна.

По определению $f(x)$ есть α_1-номер слова, имеющего α-номер, равный x. Но тогда ясно, что число

$$f^*(x) = \mu_y(x \dot{-} f(y) = 0) \qquad (7)$$

есть α-номер слова, имеющего α_1-номер, равный x, при условии, что такое слово существует. Если же такого слова нет, то формула (7) все же дает некоторое значение для $f^*(x)$. Поскольку $f^*(x) \leqslant x$, то в силу п. 3.1 заключаем, что функция $f^*(x)$ примитивно рекурсивна.

Перейдем теперь к доказательству первого утверждения теоремы. Рассмотрим какое-нибудь множество $\mathfrak{M}$ слов из $\mathfrak{S}_A$, примитивно рекурсивное в $\boldsymbol{A}_1$. Это означает, что совокупность α_1-номеров слов из $\mathfrak{M}$ совпадает с совокупностью всех решений уравнения вида $v(x) = 0$, где $v(x)$ — подходящая примитивно рекурсивная числовая функция. Но тогда совокупность всех α-номеров слов из $\mathfrak{M}$ будет совпадать с совокупностью решений уравнения $v(f(x)) = 0$, левая часть которого является примитивно рекурсивной функцией от x. Следовательно, множество $\mathfrak{M}$ примитивно рекурсивно в нумерации α.

Обратно, пусть множество $\mathfrak{M}$ примитивно рекурсивно в нумерации α и совокупность решений уравнения $v(x) = 0$ теперь является совокупностью α-номеров слов из $\mathfrak{M}$. Обозначим через $\mathfrak{N}$ множество тех слов из $\mathfrak{S}_{A_1}$, α_1-номер y которых удовлетворяет уравнению

$$v(f^*(y)) = 0.$$

Множество $\mathfrak{N}$ примитивно рекурсивное в нумерации α_1 и

$$\mathfrak{M} = \mathfrak{N} \cap \mathfrak{S}_A.$$

Поэтому остается доказать лишь примитивную рекурсивность $\mathfrak{S}_A$ в нумерации α_1. Но совокупность α_1-номеров слов из $\mathfrak{S}_A$ совпадает с совокупностью всех значений функции $f(x)$. Так как эта функция примитивно рекурсивная и удовлетворяет условию $f(x) \geqslant x$, то по теореме 3 из п. 4.1 совокупность значений f примитивно рекурсивная, что и требовалось.

Итак, примитивная рекурсивность множества слов $\mathfrak{M}$ в нумерации α_1 равносильна примитивной рекурсивности $\mathfrak{M}$ в нумерации α. Остальные утверждения теоремы 1 доказываются тем же методом.

11.2. Основные словарные операторы. По аналогии с основными вычислимыми операциями над числовыми функциями можно определить операции подстановки, рекурсии и минимизации, производимые над словарными функциями. При помощи этих «словарных операторов» легко и без каких-либо вычислений показывается примитивная рекурсивность всех наиболее употребительных словарных функций.

Пусть алфавит $\boldsymbol{A}$ состоит из букв $a_1, \ldots, a_p$. Словарные функции $\boldsymbol{O}$, $\boldsymbol{S}_i$, $\boldsymbol{I}^n_m$ $(i = 1, \ldots, p;\ m \leqslant n;\ m, n = 1, 2, \ldots)$, определенные в алфавите $\boldsymbol{A}$ равенствами

$$\boldsymbol{O}(\mathfrak{x}) = \Lambda;$$

$$\boldsymbol{S}_i(\mathfrak{x}) = \mathfrak{x}a_i;$$

$$\boldsymbol{I}^n_m(\mathfrak{x}_1, \mathfrak{x}_2, \ldots, \mathfrak{x}_n) = \mathfrak{x}_m,$$

условимся называть *простейшими*. Из п. 11.1 непосредственно следует, что простейшие словарные функции примитивно рекурсивные.

Пусть в алфавите A заданы частичная словарная функция G от n переменных и частичные словарные функции $G_1, \ldots, G_n$, каждая из которых зависит от одного и того же числа m переменных. Символом $S^{n+1}(G; G_1, \ldots, G_n)$ обозначаем частичную словарную функцию F от m переменных, определяемую обычной формулой

$$F(\mathfrak{x}_1, \ldots, \mathfrak{x}_m) = G(G_1(\mathfrak{x}_1, \ldots, \mathfrak{x}_m), \ldots, G_n(\mathfrak{x}_1, \ldots, \mathfrak{x}_m)). \quad (1)$$

Обозначим через $g, g_1, \ldots, g_n, f$ представляющие функции (п. 11.1) для словарных функций $G, G_1, \ldots, G_n, F$. Из (1) непосредственно следует, что для любых натуральных $x_1, \ldots, x_m$ имеем

$$f(x_1, \ldots, x_m) = g(g_1(x_1, \ldots, x_m), \ldots, g_n(x_1, \ldots, x_m)),$$

то есть представляющая функция для суперпозиции словарных функций равна суперпозиции представляющих функций этих словарных функций. В частности, суперпозиция примитивно рекурсивных словарных функций есть примитивно рекурсивная словарная функция.

Рассмотрим теперь какие-нибудь частичные словарные функции G и $H_1, \ldots, H_p$ соответственно от n и от $n+2$ переменных, заданные в алфавите A. Результатом операции словарной примитивной рекурсии над функциями G, $H_1, \ldots, H_p$ условимся называть $(n+1)$-местную частичную словарную функцию F, которая для любых слов $\mathfrak{x}_1, \ldots, \mathfrak{x}_n, \mathfrak{y}$ в алфавите A удовлетворяет равенствам

$$\begin{aligned} &F(\mathfrak{x}_1, \ldots, \mathfrak{x}_n, \Lambda) = G(\mathfrak{x}_1, \ldots, \mathfrak{x}_n), \\ &F(\mathfrak{x}_1, \ldots, \mathfrak{x}_n, \mathfrak{y}a_1) = H_1(\mathfrak{x}_1, \ldots, \mathfrak{x}_n, \mathfrak{y}, F(\mathfrak{x}_1, \ldots, \mathfrak{x}_n, \mathfrak{y})), \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2) \\ &F(\mathfrak{x}_1, \ldots, \mathfrak{x}_n, \mathfrak{y}a_p) = H_p(\mathfrak{x}_1, \ldots, \mathfrak{x}_n, \mathfrak{y}, F(\mathfrak{x}_1, \ldots, \mathfrak{x}_n, \mathfrak{y})). \end{aligned}$$

Легко убедиться, что для каждых $G, H_1, \ldots, H_p$ функция F существует и единственна.

Теорема 1. *Пусть частичная словарная функция F получается посредством словарной рекурсии* (2) *из частичных словарных функций $G, H_1, \ldots, H_p$. Тогда частичная числовая функция f, представляющая функцию F, может быть получена из функций $g, h_1, \ldots, h_p$, представляющих $G, H_1, \ldots, H_p$, и простейших числовых функций конечным числом операций подстановки и примитивной рекурсии.*

В частности, если заданные словарные функции G, $H_1, \ldots, H_p$ примитивно рекурсивны, то получающаяся из них с помощью словарной рекурсии функция F также примитивно рекурсивна.

В самом деле, из (2) для любых натуральных $x_1, \ldots, x_n, y$ получаем

$$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n), \qquad (3)$$

$$f(x_1, \ldots, x_n, py+i) = \\ = h_i(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y)). \qquad (4)$$

Равенство (4) можно представить в виде

$$f(x_1, \ldots, x_n, z) = \\ = \begin{cases} h_i(x_1, \ldots, x_n, [z/p], f(x_1, \ldots, x_n, [z/p])), \\ \qquad \text{если } z \dot{-} p[z/p] = i = 1, 2, \ldots, p-1, \\ h_p(x_1, \ldots, x_n, [z/p] \dot{-} 1, f(x_1, \ldots, x_n, [z/p] \dot{-} 1)), \\ \qquad \text{если } z \dot{-} p[z/p] = 0, \end{cases}$$

откуда

$$f(x_1, \ldots, x_n, z) = \sum_{\lambda=1}^{p-1} h_\lambda(x_1, \ldots, x_n, [z/p], f(x_1, \ldots \\ \ldots, x_n, [z/p]))\overline{\text{sg}}\,|\lambda - (z \dot{-} p[z/p])| + h_p(x_1, \ldots, x_n, [z/p] \dot{-} 1, \\ f(x_1, \ldots, x_n, [z/p] \dot{-} 1))\,\overline{\text{sg}}\,|z \dot{-} p[z/p]|.$$

Равенство (3) и последнее соотношение составляют в совокупности схему возвратной рекурсии для функции f, рассмотренную в п. 3.2. Поэтому согласно доказанной в п. 3.2 теореме функция f может быть получена из заданных функций g, $h_1, \ldots, h_p$ и простейших функций операциями подстановки и примитивной рекурсии.

Рассмотрим, наконец, операцию минимизации для словарных функций. Пусть $F(\mathfrak{x}, \mathfrak{y})$ — некоторая частичная словарная функция, заданная в алфавите $A = \{a_1, \ldots, a_p\}$. Если бы мы захотели определить значение выражения

$$\mu\mathfrak{y}(F(\mathfrak{x}, \mathfrak{y}) = \Lambda) \qquad (5)$$

так же, как для числовых функций, то пришлось бы говорить о *наименьшем* слове $\mathfrak{y}$, удовлетворяющем условию

$F(\mathfrak{x}, \mathfrak{y}) = \Lambda$, то есть пришлось бы пользоваться *нумерацией* всех слов в алфавите A, что не всегда желательно. Поэтому вместо одной операции минимизации (5) мы введем операцию минимизации по каждой букве алфавита a_i и будем через

$$\mu a_i^n (F(\mathfrak{x}, a_i^n) = \Lambda)$$

обозначать такое слово $a_i^n = a_i a_i \ldots a_i$ (n букв), что $F(\mathfrak{x}, a_i^n) = \Lambda$ и $F(\mathfrak{x}, a_i^s)$ определено и отлично от Λ для $s < n$.

Теорема 2. *Пусть $f(x, y)$ — представляющая частичная функция для частичной словарной функции $F(\mathfrak{x}, \mathfrak{y})$, определенной в алфавите $A = \{a_1, \ldots, a_p\}$. Тогда представляющая функция $g(x)$ для частичной словарной функции $G(\mathfrak{x})$, возникающей из F словарной минимизацией*

$$G(\mathfrak{x}) = \mu a_i^n (F(\mathfrak{x}, a_i^n) = \Lambda),$$

может быть получена из f и примитивно рекурсивных функций операциями подстановки и минимизации. В частности, если словарная функция $F(\mathfrak{x}, \mathfrak{y})$ частично рекурсивна, то и функция $G(\mathfrak{x})$ также частично рекурсивна.

Действительно, по условию имеем

$$g(x) = c\mu a_i^n (f(x, ca_i^n) = \boldsymbol{h}(\mu_t (f(x, h(t)) = 0))),$$

где

$$h(t) = ca_i^t = i + ip + \ldots + ip^{t-1}$$

— числовая примитивно рекурсивная функция.

Теперь мы с помощью теоремы 1 докажем примитивную рекурсивность некоторых часто встречающихся словарных функций. Начнем с функции $F(\mathfrak{x}, \mathfrak{y}) = \mathfrak{x}\mathfrak{y}$. Так как

$$\mathfrak{x}\Lambda = \mathfrak{x} = \boldsymbol{I}_1^1(\mathfrak{x}),$$

$$\mathfrak{x} \cdot \mathfrak{y}a_i = \mathfrak{x}\mathfrak{y} \cdot a_i = \boldsymbol{S}_i(\mathfrak{x}\mathfrak{y}) \quad (i = 1, \ldots, p)$$

и функции $\boldsymbol{I}_1^1$, $\boldsymbol{S}_i$ примитивно рекурсивные, то функция $\mathfrak{x}\mathfrak{y}$ в силу теоремы 1 также примитивно рекурсивна.

Пусть $\mathfrak{x}^{\sim}$ — обращение слова $\mathfrak{x}$, то есть $\mathfrak{x}^{\sim}$ есть слово, полученное из $\mathfrak{x}$ записью всех его букв в обратном порядке. Например, $(a_1a_2a_3)^{\sim} = a_3a_2a_1$. Из соотношений

$$\Lambda^{\sim} = \Lambda, \ (\mathfrak{x}a_i)^{\sim} = a_i\mathfrak{x}^{\sim} = \boldsymbol{S}_i\boldsymbol{O}(\mathfrak{x})\,\mathfrak{x}^{\sim}$$

видим, что $\mathfrak{x}^{\sim}$ есть примитивно рекурсивная функция от $\mathfrak{x}$.

Следствие. *Если словарные функции* $G, H_1, \ldots, H_p$ *примитивно рекурсивны и*

$$F_1(\mathfrak{x}_1, \ldots, \mathfrak{x}_n, \Lambda) = G(\mathfrak{x}_1, \ldots, \mathfrak{x}_n),$$

$$F_1(\mathfrak{x}_1, \ldots, \mathfrak{x}_n, a_i\mathfrak{y}) = H_i(\mathfrak{x}_1, \ldots, \mathfrak{x}_n, \mathfrak{y}, F_1(\mathfrak{x}_1, \ldots, \mathfrak{x}_n \mathfrak{y}))$$
$$(i = 1, \ldots, p),$$

то словарная функция F_1 *также примитивно рекурсивна.*

В самом деле, из этих соотношений следует, что функция

$$F(\mathfrak{x}_1, \ldots, \mathfrak{x}_n, \mathfrak{y}) = F_1(\mathfrak{x}_1, \ldots, \mathfrak{x}_n, \mathfrak{y}^{\sim})$$

удовлетворяет соотношениям (2) обычной словарной рекурсии и потому F примитивно рекурсивна, а вместе с нею будет примитивно рекурсивной и функция

$$F_1(\mathfrak{x}_1, \ldots, \mathfrak{x}_n, \mathfrak{y}) = F(\mathfrak{x}_1, \ldots, \mathfrak{x}_n, \mathfrak{y}^{\sim}).$$

По аналогии с числовой функцией $\operatorname{sg} x$ введем словарную функцию $\operatorname{Sg}_{\mathfrak{x}} \mathfrak{y}$, полагая

$$\operatorname{Sg}_{\mathfrak{x}} \mathfrak{y} = \begin{cases} \Lambda, & \mathfrak{y} = \Lambda, \\ \mathfrak{x}, & \mathfrak{y} \neq \Lambda. \end{cases}$$

Из соотношений

$$\operatorname{Sg}_{\mathfrak{x}} \Lambda = \Lambda, \quad \operatorname{Sg}_{\mathfrak{x}} \mathfrak{y} a_i = \mathfrak{x}$$

следует, что функция Sg примитивно рекурсивна.

Обозначим через $\mathfrak{x} - \mathfrak{y}$ решение $\mathfrak{z}$ уравнения $\mathfrak{x} = \mathfrak{z}\mathfrak{y}$. Если указанное уравнение решения не имеет, то значение выражения $\mathfrak{x} - \mathfrak{y}$ будем считать неопределенным. По аналогии с числовой разностью $\mathfrak{x} \dot{-} \mathfrak{y}$ полагаем

$$\mathfrak{x} \dot{-} \mathfrak{y} = \begin{cases} \mathfrak{x} - \mathfrak{y}, & \text{если } \mathfrak{x} - \mathfrak{y} \text{ определено}, \\ \Lambda, & \text{если } \mathfrak{x} - \mathfrak{y} \text{ не определено}. \end{cases}$$

Одноместная функция $\mathfrak{x} \dot{-} a_i$ заведомо примитивно рекурсивная, так как удовлетворяет соотношениям

$$\Lambda \dot{-} a_i = \Lambda, \quad \mathfrak{x} a_i \dot{-} a_i = \mathfrak{x}, \quad \mathfrak{x} a_j \dot{-} a_i = \Lambda \quad (j \neq i).$$

Из тождеств

$$\mathfrak{x} \dot{-} \Lambda = \mathfrak{x}, \quad \mathfrak{x} \dot{-} a_i\mathfrak{y} = (\mathfrak{x} \dot{-} \mathfrak{y}) \dot{-} a_i$$

видно, что функция $\mathfrak{x} \dot{-} \mathfrak{y}$ примитивно рекурсивна.

Мы хотим теперь показать, что функция

$$\Delta(\mathfrak{x}, \mathfrak{y}) = \begin{cases} \mathfrak{x} - \mathfrak{y}, & \textit{если } \mathfrak{x} - \mathfrak{y} \textit{ существует} \\ \mathfrak{x}, & \textit{если } \mathfrak{x} - \mathfrak{y} \textit{ не существует} \end{cases}$$

также примитивно рекурсивна. Для этого введем функции

$$\boldsymbol{D}_i(\mathfrak{x}) = \begin{cases} \Lambda, & \textit{если } \mathfrak{x} - a_i \textit{ существует}, \\ a_i, & \textit{если } \mathfrak{x} - a_i \textit{ не существует}. \end{cases}$$

Соотношения

$$\boldsymbol{D}_i(\Lambda) = a_i, \quad \boldsymbol{D}_i(\mathfrak{x}a_i) = \Lambda, \quad \boldsymbol{D}_i(\mathfrak{x}a_j) = a_i \ (j \neq i)$$

показывают, что функции $\boldsymbol{D}_i(\mathfrak{x})$ примитивно рекурсивны. Введем еще функцию $\boldsymbol{D}(\mathfrak{x}, \mathfrak{y})$, полагая по определению

$$\boldsymbol{D}(\mathfrak{x}, \Lambda) = \Lambda, \ \boldsymbol{D}(\mathfrak{x}, a_i\mathfrak{y}) = \boldsymbol{D}(\mathfrak{x}, \mathfrak{y})\, \boldsymbol{D}_i(\mathfrak{x} \dot{-} \mathfrak{y}).$$

Из этих равенств непосредственно видно, что

$$\boldsymbol{D}(\mathfrak{x}, \mathfrak{y}) = \Lambda \Leftrightarrow \mathfrak{x} - \mathfrak{y} \textit{ существует}.$$

Но в таком случае

$$\mathrm{Sg}_{\mathfrak{x}}\boldsymbol{D}(\mathfrak{x}, \mathfrak{y}) = \begin{cases} \Lambda, & \text{если } \mathfrak{x} - \mathfrak{y} \text{ существует}, \\ \mathfrak{x}, & \text{если } \mathfrak{x} - \mathfrak{y} \text{ не существует}. \end{cases}$$

Следовательно,

$$\Delta(\mathfrak{x}, \mathfrak{y}) = (\mathfrak{x} \dot{-} \mathfrak{y})\, \mathrm{Sg}_{\mathfrak{x}}\, \boldsymbol{D}(\mathfrak{x}, \mathfrak{y})$$

и потому функция Δ примитивно рекурсивна.

Полагаем по определению

$$\boldsymbol{W}_n(\mathfrak{x}_1, \ldots, \mathfrak{x}_{n+1}; \mathfrak{y}_1, \ldots, \mathfrak{y}_n) =$$

$$= \begin{cases} \mathfrak{x}_1, & \textit{если } \mathfrak{y}_1 = \Lambda, \\ \mathfrak{x}_2, & \textit{если } \mathfrak{y}_1 \neq \Lambda, \ \mathfrak{y}_2 = \Lambda, \\ \ldots & \ldots \\ \mathfrak{x}_n, & \textit{если } \mathfrak{y}_1 \neq \Lambda, \mathfrak{y}_2 \neq \Lambda, \ldots, \mathfrak{y}_n = \Lambda, \\ \mathfrak{x}_{n+1}, & \textit{если } \mathfrak{y}_1 \neq \Lambda, \mathfrak{y}_2 \neq \Lambda, \ldots, \mathfrak{y}_n \neq \Lambda. \end{cases}$$

Так как

$$\boldsymbol{W}_1(\mathfrak{x}_1, \mathfrak{x}_2; \Lambda) = \mathfrak{x}_1, \quad \boldsymbol{W}_1(\mathfrak{x}_1, \mathfrak{x}_2; \mathfrak{y}a_i) = \mathfrak{x}_2,$$

то функция $\boldsymbol{W}_1$ примитивно рекурсивна. Предполагая теперь известным, что для некоторого n функция $\boldsymbol{W}_n$

примитивно рекурсивна, из соотношений

$$\boldsymbol{W}_{n+1}(\mathfrak{x}_1, \ldots, \mathfrak{x}_{n+2}; \Lambda, \mathfrak{y}_2, \ldots, \mathfrak{y}_{n+1}) = \mathfrak{x}_1,$$
$$\boldsymbol{W}_{n+1}(\mathfrak{x}_1, \ldots, \mathfrak{x}_{n+2}; \mathfrak{y}_1 a_i, \mathfrak{y}_2, \ldots, \mathfrak{y}_{n+1}) = \\ = \boldsymbol{W}_n(\mathfrak{x}_2, \ldots, \mathfrak{x}_{n+2}; \mathfrak{y}_2, \ldots, \mathfrak{y}_{n+1})$$

видим, что функция $\boldsymbol{W}_{n+1}$ также примитивно рекурсивна.

Рассмотрим функцию $\boldsymbol{E}$, определенную схемой

$$\boldsymbol{E}(\mathfrak{x}, \mathfrak{y}) = \begin{cases} \Lambda, & \textit{если } \mathfrak{x} \in \mathfrak{y}, \\ \mathfrak{x}, & \textit{если } \mathfrak{x} \notin \mathfrak{y}, \end{cases}$$

где $\mathfrak{x} \in \mathfrak{y}$ *означает, что слово* $\mathfrak{x}$ *есть подслово в* $\mathfrak{y}$. Согласно определению имеем

$$\boldsymbol{E}(\mathfrak{x}, \Lambda) = \mathfrak{x}, \tag{6}$$

$$\boldsymbol{E}(\mathfrak{x}, \mathfrak{y}a_i) = \begin{cases} \Lambda, & \text{если } \mathfrak{x} \in \mathfrak{y}, \\ \Lambda, & \text{если } \mathfrak{x} \notin \mathfrak{y},\ \mathfrak{x} \in \mathfrak{y}a_i, \\ \mathfrak{x}, & \text{если } \mathfrak{x} \notin \mathfrak{y},\ \mathfrak{x} \notin \mathfrak{y}a_i. \end{cases} \tag{7}$$

Но условие $\mathfrak{x} \in \mathfrak{y}$ равносильно равенству $\boldsymbol{E}(\mathfrak{x}, \mathfrak{y}) = \Lambda$, а условия $\mathfrak{x} \notin \mathfrak{y}$, $\mathfrak{x} \in \mathfrak{y}a_i$ равносильны соотношениям $\boldsymbol{E}(\mathfrak{x}, \mathfrak{y}) \neq \neq \Lambda$, $\boldsymbol{D}(\mathfrak{y}a_i, \mathfrak{x}) = \Lambda$. Поэтому из (7) вытекает

$$\boldsymbol{E}(\mathfrak{x}, \mathfrak{y}a_i) = \boldsymbol{W}_2(\Lambda, \Lambda, \mathfrak{x}; \boldsymbol{E}(\mathfrak{x}, \mathfrak{y}), \boldsymbol{D}(\mathfrak{y}a_i, \mathfrak{x})). \tag{8}$$

Равенства (6), (8) показывают, что функция $\boldsymbol{E}(\mathfrak{x}, \mathfrak{y})$ примитивно рекурсивна.

Обозначим через Sb $(\mathfrak{x}; \mathfrak{y}, \mathfrak{z})$ результат подстановки слова $\mathfrak{z}$ в слово $\mathfrak{x}$ вместо первого вхождения $\mathfrak{y}$. В частности, полагаем

$$\mathrm{Sb}(\Lambda; \mathfrak{y}, \mathfrak{z}) = \Lambda \tag{9}$$

и

$$\mathrm{Sb}(\mathfrak{x}; \mathfrak{y}, \mathfrak{z}) = \mathfrak{x},$$

если $\mathfrak{y}$ в $\mathfrak{x}$ не входит. Ясно, что

$$\mathrm{Sb}(\mathfrak{x}a_i; \mathfrak{y}, \mathfrak{z}) = \begin{cases} \mathrm{Sb}(\mathfrak{x}; \mathfrak{y}, \mathfrak{z})\, a_i, & \text{если } \mathfrak{y} \in \mathfrak{x}, \\ \Delta(\mathfrak{x}a_i, \mathfrak{y})\, \mathfrak{z}, & \text{если } \mathfrak{y} \notin \mathfrak{x},\ \mathfrak{y} \in \mathfrak{x}a_i, \\ \mathfrak{x}a_i, & \text{если } \mathfrak{y} \notin \mathfrak{x},\ \mathfrak{y} \notin \mathfrak{x}a_i. \end{cases} \tag{10}$$

Соотношения (10) можно переписать в виде тождества

$$\mathrm{Sb}(\mathfrak{x}a_i; \mathfrak{y}, \mathfrak{z}) = \\ = \boldsymbol{W}_2(\mathrm{Sb}(\mathfrak{x}; \mathfrak{y}, \mathfrak{z})\, a_i,\ \Delta(\mathfrak{x}a_i, \mathfrak{y})\, \mathfrak{z},\ \mathfrak{x}a_i;\ \boldsymbol{E}(\mathfrak{y}, \mathfrak{x}),\ \boldsymbol{E}(\mathfrak{y}, \mathfrak{x}a_i)).$$

Это тождество вместе с равенством (9) составляют схему словарной рекурсии для функции Sb. Так как исходные функции примитивно рекурсивные, то и функция Sb примитивно рекурсивна.

Наконец, обозначим через $\mathrm{Sb}_a(\mathfrak{x}, \mathfrak{y})$ результат подстановки в слово $\mathfrak{x}$ слова $\mathfrak{y}$ вместо каждого вхождения в $\mathfrak{x}$ буквы a. Например, $\mathrm{Sb}_a(\mathfrak{x}, \Lambda)$ есть результат вычеркивания из $\mathfrak{x}$ всех вхождений буквы a. Далее, $\mathrm{Sb}_a(\mathfrak{x}, \Lambda) = \mathfrak{x}$ равносильно условию, что $\mathfrak{x}$ вообще не содержит буквы a и т. д. Очевидные тождества

$$\mathrm{Sb}_{a_i}(\Lambda, \mathfrak{y}) = \Lambda,$$
$$\mathrm{Sb}_{a_i}(\mathfrak{x}a_j, \mathfrak{y}) = \mathrm{Sb}_{a_i}(\mathfrak{x}, \mathfrak{y})\, a_j \quad (j \neq i),$$
$$\mathrm{Sb}_{a_i}(\mathfrak{x}a_i, \mathfrak{y}) = \mathrm{Sb}_{a_i}(\mathfrak{x}, \mathfrak{y})\, \mathfrak{y}$$

показывают, что все функции $\mathrm{Sb}_{a_i}(\mathfrak{x}, \mathfrak{y})$ являются примитивно рекурсивными.

11.3. Прямое определение класса частично рекурсивных словарных функций. Как уже отмечалось, основной недостаток введенных в п. 11.1 определений примитивной и частичной рекурсивности *словарных* функций состоит в том, что эти определения зависят от нумерации слов. Рассмотренные в п. 11.2 операции подстановки, словарной рекурсии и словарной минимизации свободны от этого недостатка и потому могут быть использованы для того, чтобы получить новые, инвариантные определения указанных понятий. Искомые определения содержатся в следующих теоремах.

Теорема 1. *Для того чтобы определенная в алфавите* **A** *словарная функция* $F(\mathfrak{x}_1, \ldots, \mathfrak{x}_n)$ *была примитивно рекурсивной, необходимо и достаточно, чтобы она могла быть получена из простейших словарных функций* $\boldsymbol{O}$, $\boldsymbol{S}$, $\boldsymbol{I}^s_m$ *конечным числом словарных рекурсий и подстановок.*

Теорема 2. *Для того чтобы заданная в алфавите* **A** *частичная словарная функция* $F(\mathfrak{x}_1, \ldots, \mathfrak{x}_n)$ *была частично рекурсивной, необходимо и достаточно, чтобы она могла быть получена из простейших словарных функций* $\boldsymbol{O}$, $\boldsymbol{S}$, $\boldsymbol{I}^s_m$ *конечным числом операций подстановки, словарной рекурсии и словарной минимизации.*

Для полноты приведем доказательства этих теорем. В п. 11.2 установлено, что операции подстановки и словарной рекурсии, примененные к примитивно рекурсивным функциям, дают функции примитивно рекурсивные и что простейшие словарные функции примитивно рекурсивны. Тем самым установлена достаточность в теореме 1. В п. 11.2 также доказано, что операция словарной минимизации, примененная к словарной частично рекурсивной функции, дает частично рекурсивную функцию. Поэтому достаточность в теореме 2 также установлена. Для доказательства необходимости в теоремах 1, 2 мы предварительно выведем несколько

лемм. Условимся словарную функцию называть *словарно примитивно рекурсивной*, если она может быть получена из простейших словарных функций операциями подстановки и словарной рекурсии.

Введем еще обозначения

$$\nu(n) = a_1^n = a_1 a_1 \ldots a_1, \ \nu(0) = a_1^0 = \Lambda,$$

где a_1 — какая-либо буква заданного алфавита.

Лемма 1. *Для каждой числовой примитивно рекурсивной функции $f(x_1, \ldots, x_n)$ существует словарно примитивно рекурсивная функция $F(\mathfrak{x}_1, \ldots, \mathfrak{x}_n)$, удовлетворяющая тождеству*

$$F(\nu x_1, \ldots, \nu x_n) = \nu f(x_1, \ldots, x_n). \tag{1}$$

Доказательство проведем по числу k операций подстановки и примитивной рекурсии, которые следует выполнить, чтобы получить f из простейших числовых функций o, s, I_i^q. Если $k = 0$ и, значит, f совпадает с одной из функций o, s, I_i^q, то словарная функция F, удовлетворяющая (1), строится очевидным образом.

Пусть $k > 0$, $f = S^{m+1}(h, h_1, \ldots, h_m)$ и пусть $H, H_1, \ldots, H_m$ — словарные функции, связанные с функциями $h, h_1, \ldots, h_m$ соотношениями вида (1). Тогда словарная функция $F = S^{m+1}(H, H_1, \ldots, H_m)$ будет связана с f соотношением (1).

Наконец, пусть $f = R(g, h)$ и пусть G, H — словарные функции, связанные с g, h соотношениями типа (1). Таким образом,

$$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n),$$
$$f(x_1, \ldots, x_n, y+1) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y))$$

и в силу (1)

$$\nu f(x_1, \ldots, x_n, 0) = G(\nu x_1, \ldots, \nu x_n),$$
$$\nu f(x_1, \ldots, x_n, y+1) = H(\nu x_1, \ldots, \nu x_n, \nu y, \nu f(x_1, \ldots, x_n, y)).$$

Но тогда $F(\mathfrak{x}_1, \ldots, \mathfrak{x}_n, \mathfrak{y})$, определенная рекурсией

$$F(\mathfrak{x}_1, \ldots, \mathfrak{x}_n, 0) = G(\mathfrak{x}_1, \ldots, \mathfrak{x}_n),$$
$$F(\mathfrak{x}_1, \ldots, \mathfrak{x}_n, \mathfrak{y}a_1) = H(\mathfrak{x}_1, \ldots, \mathfrak{x}_n, \mathfrak{y}, F(\mathfrak{x}_1, \ldots, \mathfrak{x}_n, \mathfrak{y})),$$

будет, как легко видеть, связана с f условием (1).

Лемма 2. *Существует словарно примитивно рекурсивная функция $T(\mathfrak{x})$, удовлетворяющая соотношению*

$$T(\nu(n)) = \alpha n\ \text{*}). \tag{2}$$

Действительно, обозначая через $G(\mathfrak{x})$ слово, номер которого на 1 больше номера $\mathfrak{x}$, из формулы (1) п. 11.1 видим, что

$$G(\Lambda) = a_1,$$
$$G(\mathfrak{x}a_i) = \mathfrak{x}a_{i+1} \qquad (i = 1, \ldots, p-1),$$
$$G(\mathfrak{x}a_p) = G(\mathfrak{x})\, a_1$$

и потому функция G словарно примитивно рекурсивна.

---

*) Как и ранее, αn означает слово, имеющее в алфавитной нумерации (п. 11.1) номер n.

Отсюда следует, что функция $T(\mathfrak{x})$, определенная рекурсией

$$T(\Lambda)=\Lambda,$$
$$T(\mathfrak{x}a_i)=T(\mathfrak{x}) \qquad (i=2, \ldots, p),$$
$$T(\mathfrak{x}a_1)=G(\mathfrak{x}),$$

также примитивно рекурсивная. Эта функция, очевидно, и удовлетворяет требованию (2).

Лемма 3. *Одноместная словарная функция* $\nu(c(\mathfrak{x}))$ *словарно примитивно рекурсивна.*

Согласно лемме 1 существуют словарно примитивно рекурсивные функии $F_i(\mathfrak{x})$, для которых

$$F_i(\nu x)=\nu c(\alpha x a_i)=\nu(px+i) \quad (x \in N).$$

Функция $\nu(c(\mathfrak{x}))$ удовлетворяет схеме рекурсии

$$\nu c(\Lambda)=\Lambda,$$
$$\nu c(\mathfrak{a}a_i)=\nu c(\alpha c(\mathfrak{a})\,a_i)=F_i(\nu c(\mathfrak{a})) \quad (i=1, \ldots, p)$$

и потому сама является примитивно рекурсивной.

Чтобы закончить доказательство теоремы 1, нам надо было показать, что каждая примитивно рекурсивная функция $G(\mathfrak{x}_1, \ldots, \mathfrak{x}_n)$ словарно примитивно рекурсивна. Пусть $f(x_1, \ldots, x_n)$ — представляющая функция для G. По условию f примитивно рекурсивна и, значит, в силу леммы 1 существует словарно примитивно рекурсивная функция F, связанная с f соотношением (1). Из соотношений (1), (2) последовательно получаем

$$G(\mathfrak{x}_1, \ldots, \mathfrak{x}_n)=\alpha f(c(\mathfrak{x}_1), \ldots, c(\mathfrak{x}_n))=$$
$$=T\nu f(c(\mathfrak{x}_1), \ldots, c(\mathfrak{x}_n))=TF(\nu c(\mathfrak{x}_1), \ldots, \nu c(\mathfrak{x}_n)).$$

Так как функции T, F, $\nu c(\mathfrak{x})$ словарно примитивно рекурсивны, то G также словарно примитивно рекурсивна, что и требовалось.

Доказательство теоремы 2 проводится аналогично, только вместо леммы 1 надо опираться на аналогичную лемму:

Лемма 4. *Для каждой числовой частично рекурсивной функции* $f(x_1, \ldots, x_n)$ *существует словарная частичная функция* $F(\mathfrak{x}_1, \ldots, \mathfrak{x}_n)$, *удовлетворяющая соотношению* (1) *и получающаяся из простейших словарных функций операциями подстановки, словарной рекурсии и словарной минимизации.*

Справедливость этой леммы устанавливается тем же способом, что и справедливость леммы 1. А именно, обозначаем через k число операций подстановки, примитивной рекурсии и минимизации, при помощи которых функция f получается из простейших числовых функций o, s, I_i^q, и ведем индукцию по k. Для $k=0$ утверждение очевидно. Пусть для функции f число k имеет заданное значение, а для всех функций f с меньшим k мы умеем строить функцию F. Если f получается из функций с меньшим k подстановкой или

примитивной рекурсией, то построение F было указано при доказательстве леммы 1. Остается рассмотреть случай, когда

$$f(x) = \mu_y (g(x, y) = 0) \tag{3}$$

и для функции g соответствующая функция G известна. Но из (1) и (3) непосредственно получаем

$$\nu f(x) = \mu a_1^n (G(\nu x, a_1^n) = \Lambda).$$

Следовательно, функция

$$F(\mathfrak{x}) = \mu a_1^n (G(\mathfrak{x}, a_1^n) = \Lambda)$$

и будет искомой функцией для f.

Дополнения и примеры

1. Доказать примитивную рекурсивность следующих словарных функций:

а) $F(\mathfrak{x})$ = начальной букве $\mathfrak{x}$;

б) $F(\mathfrak{x}, \mathfrak{y})$ = начальному отрезку слова $\mathfrak{x}$, длина которого равна длине $\mathfrak{y}$; если длина $\mathfrak{y}$ больше длины $\mathfrak{x}$, то $F(\mathfrak{x}, \mathfrak{y}) = \mathfrak{x}$;

в) $F(\mathfrak{x})$ = максимальному решению $\mathfrak{y}$ уравнения $\mathfrak{x} = \mathfrak{y}\mathfrak{z}\mathfrak{y}^{\sim}$.

2. Пусть алфавит A состоит из предметных переменных $x_1, \ldots, x_n$, функциональных переменных $f_1^{n_1}, \ldots, f_s^{n_s}$, скобок (,) и запятой. Показать, что совокупность всех термов, записываемых в алфавите A, примитивно рекурсивна.

3. Пусть алфавит A содержит не менее двух букв. Рассматривается совокупность всех бинарных отношений между словами в A. Для каждой буквы a_i из A определяем отношения ϱ_i, σ_i посредством формул

$$\mathfrak{x}\varrho_i\mathfrak{y} \Longleftrightarrow \mathfrak{x} = \mathfrak{y}a_i, \quad \mathfrak{x}\sigma_i\mathfrak{y} \Longleftrightarrow \mathfrak{x} = a_i\mathfrak{y}.$$

Композиция $\theta_1\theta_2$ произвольных бинарных отношений θ_1, θ_2 определяется обычным образом:

$$\mathfrak{x}\theta_1\theta_2\mathfrak{y} \Longleftrightarrow (\exists\mathfrak{z})(\mathfrak{x}\theta_1\mathfrak{z} \,\&\, \mathfrak{z}\theta_2\mathfrak{y}).$$

Исходя из отношений $\varrho_1, \sigma_1, \ldots, \varrho_p, \sigma_p$, строим новые отношения с помощью операций объединения, пересечения, композиции и рефлексивно-транзитивного замыкания отношения. Показать, что класс всех полученных отношений совпадает с классом всех бинарных рекурсивно перечислимых отношений (Г. С. Цейтин [113]).

§ 12. МАШИНЫ ТЬЮРИНГА

Первый важный и достаточно широкий класс алгоритмов был описан Тьюрингом и Постомв 1936—1937гг. ([101], [70]). Алгоритмы этого класса осуществляются особыми машинами, называемыми в настоящее время машинами Тьюринга—Поста или просто машинами Тьюринга.

Машины Тьюринга копируют в существенных чертах работу человека, вычисляющего по заданной программе, и часто рассматриваются в качестве математической модели для изучения функционирования человеческого мозга.

В настоящем параграфе сначала описываются машины Тьюринга — Поста и реализуемые ими алгоритмы переработки слов, а затем показывается, что класс функций, вычислимых на машинах Тьюринга, совпадает с классом частично рекурсивных функций.

12.1 Машины Тьюринга — Поста. Машина Тьюринга — Поста содержит следующие части (рис. 1).

а) Конечная лента (например, магнитная дорожка или бумажная лента типа телеграфной), разбитая на конечное число ячеек. В процессе работы машины к существующим ячейкам машина может пристраивать новые ячейки, так что лента может считаться потенциально неограниченной в обе стороны. Каждая ячейка ленты может находиться в одном из конечного множества состояний. Эти состояния мы будем обозначать символами $a_0, a_1, \ldots, a_m$ или другими символами. Совокупность их называется *внешним алфавитом* машины, а сама лента часто называется внешней памятью машины. Клетки в одном фиксированном состоянии иногда называются пустыми. Пустые состояния мы обычно будем обозначать символами a_0 или 0. В процессе работы машины ячейки ленты могут изменять свои состояния, но могут и не изменять их. Все вновь пристраиваемые ячейки пристраиваются пустыми. Лента считается направленной и ее ячейки мы будем просматривать слева направо. Таким образом, если в какой-то момент времени лента имеет r ячеек, и внешний алфавит машины состоит из символов $a_0, a_1, \ldots, a_m$, то состояние ленты полностью описывается словом

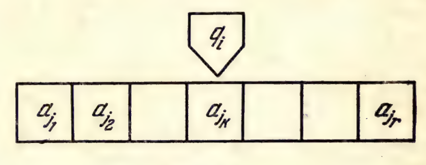


Рис. 1.

$$a_{j_1} a_{j_2} \ldots a_{j_k} \ldots a_{j_r},$$

где a_{j_1} — состояние первой (слева) ячейки, a_{j_2} — состояние второй и т. д.

б) Внутренняя память. Внутренняя память машины — это некоторое устройство, которое в каждый рассматриваемый момент находится в некотором «состоянии». Предполагается, что число возможных состояний внутренней памяти конечное и для каждой машины фиксированное. Состояния внутренней памяти мы будем обозначать символами $q_0, q_1, \ldots, q_n$ или любыми другими символами, не входящими во внешний алфавит машины. Совокупность символов, обозначающих состояния внутренней памяти, называется внутренним алфавитом машины.

Состояния внутренней памяти часто называются *внутренними состояниями* машины. Одно из этих состояний называется *заключительным* и в работе машины играет особую роль. Символ, обозначающий заключительное состояние, будет называться стоп-символом. Как правило, роль стоп-символа далее будет играть символ q_0.

в) Управляющая головка. Это некоторое устройство, которое может перемещаться вдоль ленты так, что в каждый рассматриваемый момент времени оно находится в определенной ячейке ленты. Иногда, наоборот, считают управляющую головку неподвижно связанной с машиной, а движущейся частью считают ленту. В таком случае предполагается, что в управляющую головку вводится то одна, то другая ячейка ленты. Если какая-нибудь ячейка находится в управляющей головке, то говорят также, что машина в данный момент «воспринимает» или «обозревает» эту ячейку.

г) Механическое устройство. Предполагается, что машина снабжена особым механизмом, который в зависимости от состояния воспринимаемой ячейки и состояния внутренней памяти может изменить состояние внутренней памяти и одновременно либо изменить состояние воспринимаемой ячейки, либо сдвинуть управляющую головку в соседнюю слева ячейку, либо сдвинуть управляющую головку в соседнюю справа ячейку. Если управляющая головка находится в самой правой ячейке и по ходу работы машина должна сдвинуть управляющую головку в соседнюю справа (отсутствующую) ячейку, то предполагается, что, сдвигая головку, машина одновременно пристроит недостающую ячейку в пустом состоянии.

Аналогично работает машина и в случае, когда головка воспринимает самую левую ячейку и по ходу дела машине надо сдвинуть головку еще левее.

Наконец, если в какой-то момент времени внутренняя память машины приходит в заключительное состояние q_0, то дальнейших изменений в машине не происходит и машина называется остановившейся. Может случиться, что в машине не будет происходить никаких изменений и при каком-то другом внутреннем состоянии q_i. Однако в этом случае мы не будем говорить, что машина остановилась, а будем считать, что машина продолжает работать «вечно».

По определению, *состоянием* машины Тьюринга или ее *конфигурацией* называется совокупность, образованная последовательностью $a_{j_1}, a_{j_2}, \ldots, a_{j_r}$ состояний всех ячеек ленты, состоянием внутренней памяти q_i и номером k воспринимаемой ячейки a_{j_k}.

Совокупность всех этих данных можно записать одним словом

$$a_{j_1}a_{j_2} \ldots q_i a_{j_k} \ldots a_{j_r}, \tag{1}$$

которое мы будем называть *машинным словом*, описывающим указанное состояние машины. Таким образом, каждое машинное слово содержит лишь одно вхождение символа q_i из внутреннего алфавита. Этот символ q_i может быть самым левым в машинном слове, но не может быть самым правым, так как справа от него должен помещаться символ состояния обозреваемой ячейки ленты.

Предполагается, что существуют машины Тьюринга с любыми заданными внешним и внутренним алфавитами при условии, что эти алфавиты не имеют общих символов. Предполагается также, что коль скоро заданы внешний $\{a_0, a_1, \ldots, a_m\}$ и внутренний $\{q_0, q_1, \ldots, q_n\}$ алфавиты машины, то машину можно привести в «состояние», описываемое любым произвольно заданным словом вида (1) (при условии, что символ q_i в нем не самый правый), и затем *пустить машину работать*.

Работа машины состоит в том, что она из данного «состояния» по истечении одного такта работы механического устройства переходит в следующее «состояние»,

затем от нового состояния по истечении такта работы переходит к новому состоянию и так далее.

Если данное состояние описывается машинным словом $\mathfrak{m}$, то машинное слово, описывающее следующее состояние машины, будет обозначаться через $\mathfrak{m}^{(1)}$. Полагаем, далее,

$$\mathfrak{m}^{(i+1)} = (\mathfrak{m}^{(i)})^{(1)} \qquad (i = 0, 1, 2, \ldots),$$

так что $\mathfrak{m}^{(i)}$ есть машинное слово, описывающее состояние машины, которое она приобретает через i тактов работы, отправляясь от начального состояния $\mathfrak{m}$. Утверждения $\mathfrak{m}^{(1)} = \mathfrak{b}$ и $\mathfrak{m}^{(i)} = \mathfrak{b}$, $i > 1$, мы будем записывать соответственно в виде $\mathfrak{m} \models \mathfrak{b}$ и $\mathfrak{m} \vdash \mathfrak{b}$.

Изложим теперь более точно закон построения машинного слова $\mathfrak{m}^{(1)}$ по заданному машинному слову $\mathfrak{m}$. Из описания в г) работы механического устройства следует, что машина может выполнять несколько действий. Если машина, имея внутреннее состояние q_i и воспринимая ячейку ленты в состоянии a_j, переводит внутреннюю память в состояние q_s и одновременно состояние a_j воспринимаемой ячейки заменяет состоянием a_r, то говорят, что машина *выполняет команду*

$$q_i a_j \to q_s a_r. \tag{2}$$

Если при указанных условиях машина переводит внутреннюю память в состояние q_s и передвигает управляющую головку в соседнюю справа или слева ячейку ленты, то говорят, что машина выполняет команду

$$q_i a_j \to q_s R \tag{3}$$

или соответственно команду

$$q_i a_j \to q_s L, \tag{4}$$

где R, L — особые символы.

Совокупность всех команд, которые может выполнять машина, называется ее *программой*. Так как работа машины по условию целиком определяется состоянием в данный момент ее внутренней памяти q_i и состоянием воспринимаемой ячейки a_j, то для каждых q_i, a_j ($i = 1, \ldots, n$; $j = 0, 1, \ldots, m$), программа машины должна содержать одну и только одну команду, начинающуюся словом $q_i a_j$.

Таким образом, программа машины с символами $\{a_0, a_1, \ldots, a_m\}$ и состояниями $\{q_0, q_1, \ldots, q_n\}$ содержит в точности $n(m+1)$ команд. Мы будем предполагать, что для каждого набора команд вида (2), (3), (4), содержащего для каждой пары q_i, a_j ($i = 1, \ldots, n$; $j = 0, 1, \ldots, m$) точно одну команду вида $q_i a_j \rightarrow \mathfrak{a}$, существует машина Тьюринга с этой программой.

Рассмотрим пример. Пусть внешний алфавит машины состоит из символов 0, 1, внутренний — из символов q_0, q_1, q_2, где q_0 — символ остановки, 0 — символ пустой ячейки. Пусть программа машины состоит из команд

$$\begin{aligned} &q_1 0 \rightarrow q_2 R, \quad q_2 0 \rightarrow q_0 1, \\ &q_1 1 \rightarrow q_1 R, \quad q_2 1 \rightarrow q_2 R. \end{aligned} \tag{5}$$

Посмотрим, как будут меняться состояния этой машины при тех или иных ее начальных состояниях. Вспоминая смысл команд (2), (3), (4), непосредственно находим

$$q_1 11 \models 1q_1 1 \models 11q_1 0 \models 110q_2 0 \models 110q_0 1,$$

$$q_2 101 \models 1q_2 01 \models 1q_0 11.$$

Ясно, что таким же образом и в общем случае, имея программу машины $\mathfrak{P}$ и какое-нибудь машинное слово $\mathfrak{m}$, можно строить любое число слов цепочки

$$\mathfrak{m} \models \mathfrak{m}^{(1)} \models \mathfrak{m}^{(2)} \models \ldots,$$

представляющих последовательные состояния машины.

В дальнейшем нас совсем не будут интересовать фактическое устройство машины Тьюринга, ее ленты, управляющей головки. Мы будем интересоваться лишь структурой последовательностей машинных слов, описывающих последовательные состояния, принимаемые машиной Тьюринга. С математической точки зрения машина Тьюринга — это просто определенный алгоритм для переработки машинных слов. Так как способ переработки машинных слов известен, если известна программа машины, то мы будем считать машину Тьюринга заданной, если заданы ее внешний и внутренний алфавиты, программа и указано, какие из

символов обозначают пустую ячейку и заключительное состояние.

Процесс построения слова $\mathfrak{m}^{(1)}$ по заданному машинному слову $\mathfrak{m}$ выше был описан в таких терминах, как «передвинуть управляющую головку», «изменить состояние ячейки» и т. п. Мы теперь хотим дать иное описание этого процесса, опирающееся лишь на понятие подстановки слова $\mathfrak{c}$ в слово $\mathfrak{a}$ вместо первого вхождения слова $\mathfrak{b}$ в $\mathfrak{a}$. Операция подстановки была введена в п. 11.2 и ее результат был обозначен через $\mathrm{Sb}(\mathfrak{a};\ \mathfrak{b};\ \mathfrak{c})$.

Пусть $\mathfrak{b}$, $\mathfrak{c}$ — слова в алфавите, не содержащем символа $\to$. В данном разделе формулой $\mathfrak{b} \to \mathfrak{c}$ мы будем обозначать «команду» — *в заданное слово вместо первого вхождения слова* $\mathfrak{b}$ *подставить слово* $\mathfrak{c}$, а сами формулы $\mathfrak{b} \to \mathfrak{c}$ будем называть *подстановочными командами*. Если какое-нибудь слово $\mathfrak{a}$ содержит подслово $\mathfrak{b}$, то результатом выполнения команды $\mathfrak{b} \to \mathfrak{c}$ над словом $\mathfrak{a}$ будет слово $\mathrm{Sb}(\mathfrak{a};\ \mathfrak{b};\ \mathfrak{c})$. Если же $\mathfrak{b}$ не входит в $\mathfrak{a}$, то команду $\mathfrak{a} \to \mathfrak{c}$ иногда называют *неприменимой к слову* $\mathfrak{a}$ и результат ее применения считают *неопределенным* в отличие от всегда определенной операции Sb.

Посмотрим теперь, какими подстановками из машинного слова $\mathfrak{m}$ можно получить машинное слово $\mathfrak{m}^{(1)}$. Так как иногда слово $\mathfrak{m}$ преобразуется в слово $\mathfrak{m}^{(1)}$ при помощи пристройки ячейки, то нам надо уметь отличать крайние буквы машинных слов. Для этого мы введем новые символы u, v, не входящие во внутренний и внешний алфавиты рассматриваемой машины Тьюринга $\mathfrak{T}$. Если $\mathfrak{m}$ — какое-нибудь машинное слово, то слово $u\mathfrak{m}v$ будем называть *расширенным машинным словом*, описывающим то же общее состояние машины $\mathfrak{T}$, что и слово $\mathfrak{m}$. Мы хотим сформулировать закон получения слова $u\mathfrak{m}^{(1)}v$ из слова $u\mathfrak{m}v$.

Пусть программа машины $\mathfrak{T}$ содержит команду $q_\alpha a_i \to q_\beta a_j$ и заданное расширенное машинное слово содержит подслово $q_\alpha a_i$. Ясно, что в этом случае слово $u\mathfrak{m}^{(1)}v$ получается из слова $u\mathfrak{m}v$ подстановкой $q_\alpha a_i \to q_\beta a_j$, которую мы и будем считать соответствующей команде $q_\alpha a_i \to q_\beta a_j$.

Если машинное слово $\mathfrak{m}$ содержит подслово $q_\alpha a_i$, а соответствующая машинная команда имеет вид $q_\alpha a_i \to q_\beta L$, то легко убедиться, что для получения из слова $u\mathfrak{m}v$ слова $u\mathfrak{m}^{(1)}v$ надо над словом $u\mathfrak{m}v$ выполнить ту подстановку

из совокупности

$$\begin{aligned} a_0 q_\alpha a_i &\to q_\beta a_0 a_i, \\ a_1 q_\alpha a_i &\to q_\beta a_1 a_i, \\ &\dots\dots\dots \\ a_m q_\alpha a_i &\to q_\beta a_m a_i, \\ u q_\alpha a_i &\to u q_\beta a_0 a_i, \end{aligned} \tag{6}$$

которая применима к слову $u\mathfrak{m}v$. В частности, последняя подстановка применима только тогда, когда q_α — самая левая буква в $\mathfrak{m}$, то есть когда надо пристраивать ячейку слева.

Аналогично, если машинное слово $\mathfrak{m}$ содержит подслово $q_\alpha a_i$, а соответствующая машинная команда имеет вид $q_\alpha a_i \to q_\beta R$, то чтобы из слова $u\mathfrak{m}v$ получить слово $u\mathfrak{m}^{(1)}v$, достаточно над словом $u\mathfrak{m}v$ выполнить ту из подстановочных команд

$$\begin{aligned} q_\alpha a_i a_0 &\to a_i q_\beta a_0, \\ q_\alpha a_i a_1 &\to a_i q_\beta a_1, \\ &\dots\dots\dots \\ q_\alpha a_i a_m &\to a_i q_\beta a_m, \\ q_\alpha a_i v &\to a_i q_\beta a_0 v, \end{aligned} \tag{7}$$

которая применима к слову $u\mathfrak{m}v$. Так как слово $q_\alpha a_i$ в слово $\mathfrak{m}$ входит лишь один раз, то к слову $u\mathfrak{m}v$ применима лишь одна из команд (6), (7) и потому порядки следования команд в (6) и (7) безразличны, важны лишь их совокупности.

Сформулируем теперь правило получения слова $\mathfrak{m}^{(1)}$ из машинного слова $\mathfrak{m}$ в общем виде. Пусть задана машина Тьюринга своими алфавитами $a_0, a_1, \ldots, a_m, q_0, q_1, \ldots, q_n$, выделенными символами a_0, q_0 и совокупностью машинных команд $\mathfrak{P}$. Для каждой машинной команды из $\mathfrak{P}$ строим по указанным выше правилам соответствующие подстановочные команды, совокупность которых назовем *совокупностью подстановочных команд* машины $\mathfrak{P}$. Пусть эта совокупность состоит из команд

$$\mathfrak{a}_1 \to \mathfrak{b}_1, \quad \mathfrak{a}_2 \to \mathfrak{b}_2, \; \ldots, \; \mathfrak{a}_t \to \mathfrak{b}_t. \tag{8}$$

Согласно изложенному, *для того чтобы из расширенного машинного слова* $u\mathfrak{m}v$ *получить слово* $u\mathfrak{m}^{(1)}v$, *описывающее непосредственно следующее состояние машины Тьюринга, достаточно над словом* $u\mathfrak{m}v$ *выполнить ту из подстановочных команд* (8), *которая применима к слову* $u\mathfrak{m}v$.

Если ни одна из команд (8) не применима к слову $u\mathfrak{m}v$, то $\mathfrak{m}$ содержит заключительный символ q_0 и потому слово $u\mathfrak{m}v$ описывает заключительное состояние машины.

12.2. Вычислимые функции. Рассмотрим произвольную машину Тьюринга $\mathfrak{T}$ с внешним алфавитом $a_0, a_1, \ldots \ldots, a_m$, внутренним алфавитом q_0, $q_1, \ldots, q_n$ и программой $\mathfrak{P}$. Программа $\mathfrak{P}$ позволяет преобразовывать определенным способом одни машинные слова в другие. Однако *вычисления* при помощи машины $\mathfrak{T}$ мы будем производить не над машинными словами, а над словами в *редуцированном* алфавите *) $a_1, \ldots, a_m$ машины $\mathfrak{T}$. В связи с этим введем еще несколько определений.

Пусть $\mathfrak{m}$ — какое-нибудь *машинное* слово в алфавите машины $\mathfrak{T}$. Через $\mathfrak{m}^*$ условимся обозначать слово, получающееся из $\mathfrak{m}$ вычеркиванием символа q_i и всех вхождений символа a_0. Говорят, что слово $\mathfrak{a}$, записанное в редуцированном алфавите, *перерабатывается* машиной $\mathfrak{T}$ в слово $\mathfrak{b}$, символически $\mathfrak{P}(\mathfrak{a}) = \mathfrak{b}$, если для некоторого натурального i

$$q_0 \in (q_1 a_0 \mathfrak{a})^{(i)}, \quad \mathfrak{b} = (q_1 a_0 \mathfrak{a})^{(i)*}. \tag{1}$$

Если же никакое i не удовлетворяет условию

$$q_0 \in (q_1 a_0 \mathfrak{a})^{(i)},$$

то говорят, что машина *неприменима* к слову $\mathfrak{a}$ или что результат переработки слова $\mathfrak{a}$ машиной $\mathfrak{T}$ неопределенный. В этом случае говорят также, что выражение $\mathfrak{P}(\mathfrak{a})$ имеет неопределенное значение.

В машинных терминах это означает следующее. Пусть $a_{j_1} \ldots a_{j_s}$ — какое-нибудь слово (возможно и пустое) в редуцированном алфавите $a_1, \ldots, a_m$. Берем ленту, состоящую из $s+1$ ячейки. Левую ячейку оставляем пустой, а в остальные последовательно «вписываем» символы

---

*) Символы a_0, u, v исключаются.

$a_{j_1}, \ldots, a_{j_s}$. Помещаем ленту в машину (рис. 2) так, чтобы машина воспринимала самую левую (пустую) ячейку. Приводим внутреннюю память машины в *начальное состояние* q_1 и пускаем машину в ход. В силу своего устройства машина $\mathfrak{T}$ начнет последовательно принимать состояния

$$(q_1 a_0 a_{j_1} \ldots a_{j_s}) \models (q_1 a_0 a_{j_1} \ldots a_{j_s})^{(1)} \models (q_1 a_0 a_{j_1} \ldots a_{j_s})^{(2)} \models \ldots$$

Теперь возможны два случая: либо в какой-то момент времени i внутренняя память машины придет в заключительное состояние и, значит, машинное слово, описывающее состояние машины в этот момент, будет иметь вид $\mathfrak{a}_i q_0 \mathfrak{b}_i$; либо внутренняя память никогда не придет в состояние q_0. В первом случае $\mathfrak{P}(a_{j_1} \ldots a_{j_s}) = \mathfrak{a}_i^* \mathfrak{b}_i^*$, а во втором случае $\mathfrak{P}(a_{j_1} \ldots a_{j_s})$ имеет «неопределенное» значение.

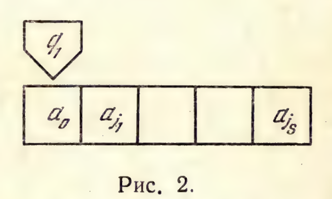


Рис. 2.

Пусть $F(\mathfrak{x})$ — какая-нибудь словарная частичная функция в редуцированном алфавите $a_1, \ldots, a_m$ машины $\mathfrak{T}$. Если для каждого слова $\mathfrak{x}$ в этом алфавите

$$F(\mathfrak{x}) = \mathfrak{P}(\mathfrak{x}),$$

то говорят, что машина $\mathfrak{T}$ *вычисляет* функцию $F(\mathfrak{x})$.

Рассмотрим, например, машину с программой (5) из предыдущего раздела. Редуцированный алфавит этой машины содержит лишь символ 1 и потому слова в этом алфавите имеют вид

$$\Lambda, \quad 1, \quad 11, \quad 111, \ldots$$

Из соотношений

$$q_1 0 \models 0 q_2 0 \models 0 q_0 1,$$

$$q_1 11 \ldots 1 \models 1 q_1 1 \ldots 1 \models \ldots \models 1 \ldots 1 q_1 0 \models$$
$$\models 1 \ldots 10 q_2 0 \models 1 \ldots 10 q_0 1$$

видим, что $\mathfrak{P}(\mathfrak{x}) = \mathfrak{x}1$.

С другой стороны, рассмотрим машину $\mathfrak{T}_1$ с алфавитами $\{0, 1\}$, $\{q_0, q_1\}$ и программой

$$q_1 0 \to q_1 R, \quad q_1 1 \to q_1 R.$$

Из соотношений

$$q_1 0 \models 0 q_1 0 \models 00 q_1 0 \models \ldots,$$

$$q_1 01 \ldots 1 \models \ldots \models 01 \ldots 1 q_1 0 \models 01 \ldots 10 q_1 0 \models \ldots$$

заключаем, что $\mathfrak{T}_1$ вычисляет нигде не определенную функцию.

Возвращаясь к машине Тьюринга с произвольными алфавитами $\{a_0, a_1, \ldots a_m\}$, $\{q_0, q_1, \ldots, q_n\}$ и программой $\mathfrak{P}$, условимся говорить, что машина $\mathfrak{T}$ *s-ку слов* $\mathfrak{x}_1, \ldots, \mathfrak{x}_s$, записанных в редуцированном алфавите $a_1, \ldots, a_m$, *перерабатывает* в слово $\mathfrak{x}$, если для подходящего натурального i

$$q_0 \in (q_1 a_0 \mathfrak{x}_1 a_0 \mathfrak{x}_2 \ldots a_0 \mathfrak{x}_s)^{(i)}, \quad \mathfrak{x} = (q_1 a_0 \mathfrak{x}_1 \ldots a_0 \mathfrak{x}_s)^{(i)*}.$$

В этом случае пишут $\mathfrak{P}(\mathfrak{x}_1, \ldots, \mathfrak{x}_s) = \mathfrak{x}$. Если для любого i

$$q_0 \notin (q_1 a_0 \mathfrak{x}_1 \ldots a_0 \mathfrak{x}_s)^{(i)},$$

то говорят, что машина $\mathfrak{T}$ *неприменима к s*-ке $\mathfrak{x}_1, \ldots, \mathfrak{x}_s$ и что выражение $\mathfrak{P}(\mathfrak{x}_1, \ldots, \mathfrak{x}_s)$ имеет *неопределенное* значение.

Говорят, что словарная частичная функция $F(\mathfrak{x}_1, \ldots, \mathfrak{x}_s)$, определенная в алфавите $\{a_1, \ldots, a_k\}$, составляющем часть редуцированного алфавита $\{a_1, \ldots, a_k, \ldots, a_m\}$ машины $\mathfrak{T}$, *вычисляется* машиной $\mathfrak{T}$, если для любой s-ки слов $\mathfrak{x}_1, \ldots, \mathfrak{x}_s$ в алфавите $\{a_1, \ldots, a_k\}$

$$F(\mathfrak{x}_1, \ldots, \mathfrak{x}_s) = \mathfrak{P}(\mathfrak{x}_1, \ldots, \mathfrak{x}_s).$$

Таким образом, с каждой машиной Тьюринга $\mathfrak{T}$, имеющей указанный внешний алфавит $\{a_0, a_1, \ldots, a_m\}$, сопоставляются частичные словарные функции $\mathfrak{P}(\mathfrak{x})$, $\mathfrak{P}(\mathfrak{x}_1, \mathfrak{x}_2), \ldots$, определенные в редуцированном алфавите $\{a_1, \ldots, a_m\}$. Остальные функции, вычислимые на машине $\mathfrak{T}$,— это ограничения функций $\mathfrak{P}(\mathfrak{x})$, $\mathfrak{P}(\mathfrak{x}_1, \mathfrak{x}_2), \ldots$ на множествах слов, записываемых в частичных алфавитах $\{a_1, \ldots, a_k\}$ $(k \leqslant m)$.

Частичная словарная функция называется *вычислимой* или *вычислимой по Тьюрингу*, если она вычислима на подходящей машине Тьюринга.

Теорема 1 (Тьюринг). *Все частичные словарные функции, вычислимые по Тьюрингу, являются частично рекурсивными.*

Так как функция, определенная в некотором алфавите и частично рекурсивная в расширенном алфавите, является частично рекурсивной и в первоначальном алфавите (п. 11.1), то для доказательства теоремы 1 достаточно показать, что функции $\mathfrak{P}(\mathfrak{x}_1, \ldots, \mathfrak{x}_s)$ $(s = 1, 2, \ldots)$ для любой машины Тьюринга $\mathfrak{T}$ являются частично рекурсивными.

Пусть $\{a_0, a_1, \ldots, a_m\}$, $\{q_0, q_1, \ldots, q_n\}$— внешний и внутренний алфавиты машины $\mathfrak{T}$ и пусть

$$\mathfrak{b}_1 \longrightarrow \mathfrak{c}_1, \quad \mathfrak{b}_2 \longrightarrow \mathfrak{c}_2, \ldots, \mathfrak{b}_t \longrightarrow \mathfrak{c}_t \tag{2}$$

— *программа подстановок* для $\mathfrak{T}$, определенная в предшествующем разделе. Символы $\mathfrak{b}_i$, $\mathfrak{c}_i$ означают подходящие слова в расширенном алфавите $\{a_0, a_1, \ldots, a_m, q_0, \ldots, q_n, u, v\}$.

Для произвольного слова $\mathfrak{a}$ в указанном объединенном алфавите мы определяем слово $\mathfrak{a}^{(1)}$ с помощью следующего правила: *в программе подстановок* (2) *ищем первую команду* $\mathfrak{b}_i \to \mathfrak{c}_i$, *левое слово которой* $\mathfrak{b}_i$ *входит в* $\mathfrak{a}$, *и затем в* $\mathfrak{a}$ *подставляем слово* $\mathfrak{c}_i$ *вместо первого вхождения слова* $\mathfrak{b}_i$. *Если ни одно из левых слов команд программы* (2) *в* $\mathfrak{a}$ *не входит, то полагаем* $\mathfrak{a}^{(1)} = \mathfrak{a}$.

Изложенное правило дословно совпадает с правилом построения слова $u\mathfrak{m}^{(1)}v$ по заданному расширенному машинному слову $u\mathfrak{m}v$, которое было сформулировано в конце предыдущего раздела. Отличие состоит лишь в том, что теперь мы применяем это правило к произвольным словам в расширенном алфавите, а не только к машинным словам. Полагая, как и в п. 12.1,

$$\mathfrak{a}^{(0)} = \mathfrak{a}, \quad \mathfrak{a}^{(\lambda+1)} = \mathfrak{a}^{(\lambda)(1)}, \qquad (\lambda = 0, 1, 2, \ldots),$$

мы видим, что выражение $\mathfrak{a}^{(i)}$ теперь имеет определенное значение для любого слова $\mathfrak{a}$ в расширенном алфавите, причем, если $\mathfrak{a}$ — машинное слово, то $\mathfrak{a}^{(i)}$ — машинное слово, отвечающее тому состоянию машины Тьюринга, в которое она придет из состояния $\mathfrak{a}$ после i тактов работы.

Вместо выражения $\mathfrak{x}^{(i)}$, зависящего от словарного аргумента $\mathfrak{x}$ и числового аргумента i, вводим функцию $V(\mathfrak{a}, \mathfrak{x})$, полагая

$$V(\mathfrak{a}, \mathfrak{x}) = \mathfrak{x}^{(\text{длина } \mathfrak{a})}$$

для любых слов $\mathfrak{a}$, $\mathfrak{x}$ в расширенном алфавите.

Функция $\mathfrak{P}(\mathfrak{x}_1, \ldots, \mathfrak{x}_s)$ определена в редуцированном алфавите. Мы продолжим эту функцию до функции $\mathfrak{P}_0(\mathfrak{x}_1, \ldots, \mathfrak{x}_s)$ в объединенном алфавите, полагая $\mathfrak{P}_0(\mathfrak{x}_1, \ldots, \mathfrak{x}_s)$ совпадающей с $\mathfrak{P}(\mathfrak{x}_1, \ldots, \mathfrak{x}_s)$ для слов $\mathfrak{x}_1, \ldots, \mathfrak{x}_s$ в редуцированном алфавите и неопределенной для остальных значений $\mathfrak{x}_1, \ldots, \mathfrak{x}_s$. Так как графики функций $\mathfrak{P}$ и $\mathfrak{P}_0$ совпадают, то частичная рекурсивность одной из этих функций влечет за собой частичную рекурсивность другой. Но сравнивая определения функций $\mathfrak{P}$ и V, непосредственно заключаем, что для слов $\mathfrak{x}_1, \ldots, \mathfrak{x}_s, \mathfrak{x}$ в расширенном алфавите равенство

$$\mathfrak{P}_0(\mathfrak{x}_1, \ldots, \mathfrak{x}_s) = \mathfrak{x} \tag{3}$$

тогда и только тогда истинно, когда *существует слово* $\mathfrak{a}$, удовлетворяющее требованиям:

$$\begin{gathered} \mathfrak{x}_1, \ldots, \mathfrak{x}_s \textit{ не содержат букв } a_0, q_0, \ldots, q_n, u, v, \\ q_0 \textit{ содержится в слове } V(\mathfrak{a}, uq_1a_0\mathfrak{x}_1 \ldots a_0\mathfrak{x}_s v), \\ \mathfrak{x} = V(\mathfrak{a}, uq_1a_0\mathfrak{x}_1 \ldots a_0\mathfrak{x}_s v)^*, \end{gathered} \tag{4}$$

где звездочка * означает операцию выбрасывания из данного слова всех вхождений символов $a_0, q_0, \ldots, q_n, u, v$.

Согласно п. 11.2

$$\begin{gathered} a_0 \notin \mathfrak{x}_i \Leftrightarrow \Lambda = E(\mathfrak{x}_i, \mathrm{Sb}_{a_0}(\mathfrak{x}_i, \Lambda)), \\ q_j \notin \mathfrak{x}_i \Leftrightarrow \Lambda = E(\mathfrak{x}_i, \mathrm{Sb}_{q_j}(\mathfrak{x}_i, \Lambda)) \\ (j = 0, \ldots, n), \\ u \notin \mathfrak{x}_i \Leftrightarrow \Lambda = E(\mathfrak{x}_i, \mathrm{Sb}_u(\mathfrak{x}_i, \Lambda)), \\ v \notin \mathfrak{x}_i \Leftrightarrow \Lambda = E(\mathfrak{x}_i, \mathrm{Sb}_v(\mathfrak{x}_i, \Lambda)), \\ q_0 \in V(\mathfrak{a}, uq_1a_0\mathfrak{x}_1 \ldots a_0\mathfrak{x}_s v) \Leftrightarrow \Lambda = E(q_0, V(\mathfrak{a}, uq_1a_0\mathfrak{x}_1 \ldots a_0\mathfrak{x}_s v)), \\ \mathfrak{x} = V(\mathfrak{a}, uq_1a_0\mathfrak{x}_1 \ldots a_0\mathfrak{x}_s v)^* \Leftrightarrow \Lambda = E(\mathfrak{x}, V(\ldots)^*)\, E(V(\ldots)^*, \mathfrak{x}), \end{gathered} \tag{5}$$

где E, Sb_a — примитивно рекурсивные словарные функции, определенные в конце п. 11.2.

Обозначая через $G(\mathfrak{x}_1, \ldots, \mathfrak{x}_s, \mathfrak{x}, \mathfrak{a})$ произведение всех выражений, стоящих справа от (правых) знаков равенств в формулах (5), мы приходим к заключению, что совокупность условий (4) равносильна равенству

$$G(\mathfrak{x}_1, \ldots, \mathfrak{x}_s, \mathfrak{x}, \mathfrak{a}) = \Lambda. \tag{6}$$

График функции $\mathfrak{P}_0$ состоит из систем $(\mathfrak{x}_1, \ldots, \mathfrak{x}_s, \mathfrak{x})$, удовлетворяющих условию (3), то есть из систем, для которых уравнение (6) имеет хотя бы одно решение $\mathfrak{a}$.

Допустим, что функция $V(\mathfrak{a}, \mathfrak{x})$ примитивно рекурсивна. Тогда левая часть уравнения (6) будет также примитивно рекурсивной функцией, график функции $\mathfrak{P}_0$ будет рекурсивно перечислимым, а сама функция $\mathfrak{P}_0$ будет частично рекурсивной, что и требуется. Поэтому остается лишь доказать примитивную рекурсивность функции V.

Из определения функции V следует, что

$$V(\Lambda, \mathfrak{x}) = \mathfrak{x}, \tag{7}$$

$$V(\mathfrak{a}z, \mathfrak{x}) = \begin{cases} \mathrm{Sb}(V(\mathfrak{a}, \mathfrak{x}); \mathfrak{b}_1, \mathfrak{c}_1), & \text{если } \mathfrak{b}_1 \in V(\mathfrak{a}, \mathfrak{x}), \\ \mathrm{Sb}(V(\mathfrak{a}, \mathfrak{x}); \mathfrak{b}_2, \mathfrak{c}_2), & \text{если } \mathfrak{b}_1 \notin V(\mathfrak{a}, \mathfrak{x}),\ \mathfrak{b}_2 \in V(\mathfrak{a}, \mathfrak{x}), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ V(\mathfrak{a}, \mathfrak{x}), & \text{если } \mathfrak{b}_1 \notin V(\mathfrak{a}, \mathfrak{x}), \ldots, \mathfrak{b}_t \notin V(\mathfrak{a}, \mathfrak{x}), \end{cases} \tag{8}$$

где $z \in \{a_0, a_1, \ldots, q_0, \ldots, q_n, u, v\}$, Sb — знак подстановки (п. 11.2), а $\mathfrak{b}_1, \mathfrak{c}_1, \ldots, \mathfrak{b}_t, \mathfrak{c}_t$ — слова из программы подстановок (2) машины $\mathfrak{T}$.

Согласно п. 11.2 схемы (7), (8) равносильны схеме словарной индукции для $V(\mathfrak{a}, \mathfrak{x})$ с примитивно рекурсивными исходными функцией $\mathfrak{x}$ и функциями

$$H_c(\mathfrak{a}, \mathfrak{x}, \mathfrak{y}) = W(\mathrm{Sb}(\mathfrak{y}; \mathfrak{b}_1, \mathfrak{c}_1), \ldots, \mathrm{Sb}(\mathfrak{y}; \mathfrak{b}_t, \mathfrak{c}_t), \mathfrak{y}; \\ E(\mathfrak{b}_1, \mathfrak{y}), \ldots, E(\mathfrak{b}_t, \mathfrak{y})).$$

Следовательно, функция $V(\mathfrak{a}, \mathfrak{x})$ примитивно рекурсивна и теорема доказана.

Обратная теорема к доказанной теореме также верна. Однако доказательство ее требует некоторых вычислений, которые будут проведены в следующем разделе.

12.3. Синтез машин Тьюринга. Как уже говорилось, имеет место следующая важная

Теорема 1 (Тьюринг). *Для каждой частично рекурсивной словарной функции $F(\mathfrak{x}_1, \ldots, \mathfrak{x}_s)$, определенной в некотором алфавите $a_1, \ldots, a_m$, существует машина Тьюринга с символами $a_0, a_1, \ldots, a_m$ и подходящими внутренними состояниями, которая вычисляет функцию $F(\mathfrak{x}_1, \ldots, \mathfrak{x}_s)$.*

Каждая частично рекурсивная функция получается из простейших функций операциями подстановки, примитивной рекурсии и обращения. Поэтому теорема 1 будет доказана, если нам удастся решить следующие частные задачи:

1) построить машины Тьюринга, вычисляющие простейшие функции;

2) имея машины Тьюринга, вычисляющие функции F, $F_1, \ldots, F_s$, построить машину, вычисляющую функцию $F(F_1, \ldots, F_s)$;

3) имея машины Тьюринга, вычисляющие какие-то функции G, H_i, построить машину, вычисляющую функцию, возникающую из G, H_i примитивной рекурсией;

4) имея машину Тьюринга, вычисляющую всюду определенную функцию $G(\mathfrak{x}_1, \ldots, \mathfrak{x}_s, \mathfrak{x}, \mathfrak{a})$, построить машину, вычисляющую функцию F, заданную соотношением

$$F(\mathfrak{x}_1, \ldots, \mathfrak{x}_s) = \mathfrak{x} \Longleftrightarrow (\exists \mathfrak{a})(G(\mathfrak{x}_1, \ldots, \mathfrak{x}_s, \mathfrak{x}, \mathfrak{a}) = \Lambda).$$

Здесь $(\exists \mathfrak{a})$ означает выражение «существует такое слово $\mathfrak{a}$, что ...» (см. основные обозначения).

Задачи 1)—4) и будут решены ниже. Внутренние состояния искомой машины будут обозначаться символами q_0, $q_1, \ldots, q_n$, а сама машина будет задаваться своей программой команд $\mathfrak{P}$, причем иногда будут выписываться не все команды $\mathfrak{P}$, а лишь те, которые будут нужны по ходу дела. Как и в предыдущем разделе, если $\mathfrak{m}$ — машинное слово, то через $\mathfrak{m}^{(i)}$ будем обозначать машинное слово, описывающее то состояние машины, в которое она придет, исходя из состояния $\mathfrak{m}$, через i тактов работы.

Для машинных слов $\mathfrak{m}$, $\mathfrak{n}$ будем писать $\mathfrak{m} \Longrightarrow \mathfrak{n}$, если для некоторого i $\mathfrak{m}^{(i)} = \mathfrak{n}$ и в процессе перехода от состояния $\mathfrak{m}$ к состоянию $\mathfrak{n}$ машина ни разу не пристраивала ячеек слева. Мы будем писать $\mathfrak{m} \models\!\!\Longrightarrow \mathfrak{n}$, если для некоторого i $\mathfrak{m}^{(i)} = \mathfrak{n}$ и в процессе перехода от состояния $\mathfrak{m}$ к состоянию $\mathfrak{n}$ машина не пристраивала новых ячеек ни слева, ни справа.

Пусть машина $\mathfrak{T}$ имеет алфавиты $\{a_0, a_1, \ldots, a_m\}$, $\{q_0, q_1, \ldots, q_n\}$ и $F(\mathfrak{x}_1, \ldots, \mathfrak{x}_s)$ — частичная s-местная словарная функция, заданная в редуцированном алфавите $\{a_1, \ldots, a_m\}$. Мы будем говорить, что машина $\mathfrak{T}$ *правильно вычисляет* частичную функцию $F(\mathfrak{x}_1, \ldots, \mathfrak{x}_s)$, если

$$q_1 a_0 \mathfrak{x}_1 a_0 \mathfrak{x}_2 \ldots a_0 \mathfrak{x}_s \models\!\!\Longrightarrow q_0 a_0 F(\mathfrak{x}_1, \ldots, \mathfrak{x}_s) a_0 \ldots a_0$$

для любой системы слов $\mathfrak{x}_1, \ldots, \mathfrak{x}_s$ в редуцированном алфавите, принадлежащей к области определенности функции F, и если в случае, когда $F(\mathfrak{x}_1, \ldots, \mathfrak{x}_s)$ не определено, машина $\mathfrak{T}$, начав работать в состоянии $q_1 a_0 \mathfrak{x}_1 a_0 \mathfrak{x}_2 \ldots a_0 \mathfrak{x}_s$, никогда не остановится, т. е. не придет во внутреннее состояние q_0, и никогда в процессе работы не будет надстраивать ленту слева.

Сравнивая понятие правильной вычислимости функции с понятием (простой) вычислимости, введенным в п. 12.2, непосредственно видим, что из правильной вычислимости заведомо вытекает простая вычислимость. Поэтому мы лишь усилим теорему 1, если докажем, что справедлива

Теорема 2. *Для каждой частично рекурсивной словарной функции $F(\mathfrak{x}_1, \ldots, \mathfrak{x}_s)$, определенной на совокупности слов какого-нибудь алфавита $\{a_1, \ldots, a_m\}$, существует машина Тьюринга с символами $\{a_0, a_1, \ldots, a_m\}$ и подходящими внутренними состояниями, которая правильно вычисляет функцию F.*

Мы докажем эту теорему лишь для случая, когда первоначальный алфавит $\{a_1, \ldots, a_m\}$ состоит из одного символа 1. Рассуждения в случае произвольного алфавита остаются теми же самыми и лишь записи становятся более громоздкими.

Вместо символа a_0 мы будем употреблять далее символ 0. Для положительного натурального x вводим обозначения

$$1^x = 11\ldots1, \quad 0^x = 00\ldots0.$$

Дополнительно полагаем

$$0^0 = 1^0 = \Lambda,$$

где Λ — пустое слово.

На слова

$$1^0 = \Lambda,\ 1^1 = 1,\ 1^2 = 11,\ 1^3 = 111,\ \ldots$$

мы будем смотреть как на «изображения» натуральных чисел 0, 1, 2, 3, ... Соответственно этому мы будем говорить, что числовая частичная функция $f(x_1, \ldots, x_s)$ правильно вычисляется машиной $\mathfrak{T}$, если этой машиной правильно вычисляется словарная функция $F(\mathfrak{x}_1, \ldots, \mathfrak{x}_s)$,

определенная равенством

$$F(1^{x_1}, 1^{x_2}, \ldots, 1^{x_s}) = 1^{f(x_1, x_2, \ldots, x_s)}.$$

Итак, мы хотим доказать, что *для каждой числовой частично рекурсивной функции* $f(x_1, \ldots, x_s)$ *существует машина Тьюринга с внешним алфавитом* $\{0, 1\}$ *и подходящим внутренним алфавитом* $\{q_0, q_1, \ldots, q_n\}$, *которая правильно вычисляет функцию* $f(x_1, \ldots, x_s)$, *то есть для которой*

$$q_1 0 1^{x_1} 0 1^{x_2} \ldots 0 1^{x_s} \Rightarrow q_0 0 1^{f(x_1, \ldots, x_s)} 0 \ldots 0$$

при любых $x_1, \ldots, x_s$ *из области определенности функции* f, *и которая не останавливается и не надстраивает ячеек слева при запуске ее из состояния* $q_1 0 1^{x_1} 0 1^{x_2} \ldots 0 1^{x_s}$ *в случае, если значение* $f(x_1, x_2, \ldots, x_s)$ *не определено.*

Мы начнем с определения понятия композиции машин Тьюринга, играющего основную роль во всех вопросах, связанных с синтезом машин. Пусть заданы машины Тьюринга $\mathfrak{P}$, $\mathfrak{Q}$, имеющие какой-то общий внешний алфавит $\{a_0, a_1, \ldots, a_m\}$ и внутренние алфавиты $\{q_0, q_1, \ldots, q_n\}$ и соответственно $\{q_0, q'_1, \ldots, q'_t\}$. *Композитом* (или произведением) машины $\mathfrak{P}$ на машину $\mathfrak{Q}$ мы будем называть машину $\mathfrak{R}$ с тем же внешним алфавитом $\{a_0, a_1, \ldots, a_m\}$, внутренним алфавитом $\{q_0, q_1, \ldots, q_n, q_{n+1}, \ldots, q_{n+t}\}$ и программой, получающейся следующим образом. Во всех командах из $\mathfrak{P}$, содержащих заключительный символ q_0, заменяем q_0 символом q_{n+1}. Все остальные символы в командах из $\mathfrak{P}$ оставляем неизменными. В командах из $\mathfrak{Q}$, напротив, символ q_0 оставляем неизменным, но зато каждый из остальных символов q'_j $(j = 1, \ldots, t)$ заменяем символом q_{t+j}. Совокупность всех команд из $\mathfrak{P}$ и $\mathfrak{Q}$, измененных указанным способом (обозначим их соответственно $\mathfrak{P}^*$ и $\mathfrak{Q}^*$), и будет программой композита машин $\mathfrak{P}$, $\mathfrak{Q}$.

Композит машины $\mathfrak{P}$ на машину $\mathfrak{Q}$ обозначается через $\mathfrak{P}\mathfrak{Q}$ или $\mathfrak{P} * \mathfrak{Q}$. Основное свойство его выражает

Лемма 1. *Если для каких-то слов* $\mathfrak{a}_0, \mathfrak{b}_0, \mathfrak{a}_1, \mathfrak{b}_1, \mathfrak{a}_2, \mathfrak{b}_2$ *внешнего алфавита* $\{a_0, a_1, \ldots, a_m\}$

$$\mathfrak{P}\colon\ \mathfrak{a}_0 q_1 \mathfrak{b}_0 \Rightarrow \mathfrak{a}_1 q_0 \mathfrak{b}_1,$$

$$\mathfrak{Q}\colon\ \mathfrak{a}_1 q_1 \mathfrak{b}_1 \Rightarrow \mathfrak{a}_2 q_0 \mathfrak{b}_2,$$

то

$$\mathfrak{P}\mathfrak{Q}\colon\ \mathfrak{a}_0 q_1 \mathfrak{b}_0 \Rightarrow \mathfrak{a}_2 q_0 \mathfrak{b}_2.$$

Действительно, преобразуя слово $\mathfrak{a}_0 q_1 \mathfrak{b}_0$ с помощью команд из $\mathfrak{P}^*$, получим слово $\mathfrak{a}_1 q_{n+1} \mathfrak{b}_1$. Дальнейшие преобразования его с помощью команд из $\mathfrak{Q}^*$ приведут к $\mathfrak{a}_2 q_0 \mathfrak{b}_2$.

Отметим следующее очевидное положение: если программа машины содержит команду $q_i 1 \to q_i L$, то

$$01 \ldots 1 q_i 1 \Longrightarrow q_i 01 \ldots 11.$$

Аналогично, если программа машины содержит команду $q_i 1 \to q_i R$, то

$$q_i 1 \ldots 10 \Longrightarrow 1 \ldots 1 q_i 0.$$

Ниже указывается серия машин Тьюринга, преобразующих машинные слова одного заданного вида в машинные слова другого заданного вида. Программы машин пишутся столбцами. Для контроля против некоторых команд указываются те машинные слова, в которые переходит первоначально заданное машинное слово после выполнения всех предыдущих команд, включая в их число и стоящую слева команду. Выписанные ниже программы отнюдь не самые короткие, но в каком-то смысле естественные.

А) Перенос нуля: $q_1 001^x 0 \Longrightarrow q_0 01^x 00$

Имеем:
$q_1 0 \to q_2 R$ $\quad 0 q_2 01^x 0$
$q_2 0 \to q_3 1$
$q_3 1 \to q_3 R$ $\quad 011^x q_3 0$
$q_3 0 \to q_4 L$
$q_4 1 \to q_5 0$ $\quad 01^x q_5 00$
$q_5 0 \to q_6 L$
$q_6 1 \to q_6 L$ $\quad q_6 01^x 00$
$q_6 0 \to q_0 0$ $\quad q_0 01^x 00.$

Б⁻) Левый сдвиг: $01^x q_1 0 \Longrightarrow q_0 01^x 0$

Имеем:
$q_1 0 \to q_2 L$
$q_2 1 \to q_2 L$ $\quad q_2 01^x 0$
$q_2 0 \to q_0 0$ $\quad q_0 01^x 0.$

Б⁺) Правый сдвиг: $q_1 01^x 0 \Longrightarrow 01^x q_0 0.$

Программа этой машины получается из предыдущей программы заменой символа L символом R.

В) Транспозиция: $01^x q_1 01^y 0 \Longrightarrow 01^y q_0 01^x 0$.

Сначала слово $01^x q_1 01^y 0$ переводим в слово $01^x q 1^y 00$. При $y>0$ это достигается следующей очевидной программой:

$$\text{Б}^+\text{:}\ 01^x 01^y q_3 0$$

$$\begin{array}{ll} q_3 0 \to q_4 L & \\ q_4 1 \to q_5 0 & \\ q_5 0 \to q_6 L & \\ q_6 1 \to q_6 L & 01^x q_6 01^{y-1} 00 \\ q_6 0 \to q_7 1 & 01^x q_7 1^y 00. \end{array}$$

Чтобы получить слово $01^x q_7 1^y 00$ и при $y = 0$, добавляем команду

$$q_4 0 \to q_7 0 \quad 01^x q_7 1^y 00 \quad (y \geqslant 0).$$

Теперь из подслова 1^x перебрасываем в промежуток между двумя последними нулями один символ 1. Очевидно, это достигается следующей программой:

$$\begin{array}{ll} q_7 1 \to q_8 L & 01^{x-1} q_8 1 1^y 00 \\ q_8 1 \to q_9 0 & \\ q_9 0 \to q_{10} R & \\ q_{10} 1 \to q_{10} R & 01^{x-1} 01^y q_{10} 00 \\ q_{10} 0 \to q_{11} 1 & \\ q_{11} 1 \to q_{12} L & \\ q_{12} 1 \to q_{13} 0 & 01^{x-1} 01^{y-1} q_{13} 010 \\ q_{13} 0 \to q_{14} L & \\ q_{14} 1 \to q_{14} L & 01^{x-1} q_{14} 01^{y-1} 010 \\ q_{14} 0 \to q_{15} 1 & 01^{x-1} q_{15} 1^y 010. \end{array}$$

Мы достигли своей цели, но в предположении $x>0$, $y>0$.

Чтобы тот же результат получился и при $y = 0$, добавляем к записанной программе еще приказы

$$\begin{array}{ll} q_7 0 \to q_{16} 1 & \\ q_{16} 1 \to q_{17} L & \\ q_{17} 1 \to q_{15} 0 & 01^{x-1} q_{15} 1^y 010 \ (x > 0,\ y \geqslant 0). \end{array}$$

Теперь мы «зацикливаем» программу приказами

$$q_{15}1 \to q_7 1$$
$$q_{15}0 \to q_7 0 \quad 01^{x-1}q_7 1^y 010.$$

Если $x-1>0$, то, как легко видеть, слово $01^{x-1}q_7 1^y 010$ переработается в слово $01^{x-2}q_7 1^y 0110$. Если $x-2>0$, то процесс дальнейшей переработки даст слово $01^{x-3}q_7 1^y 01110$ и т. д. Через x «циклов» получим слово $0q_7 1^y 01^x 0$, которое после выполнения приказа $q_7 1 \to q_8 L$ (если $y>0$) или приказов $q_7 0 \to q_{16} 1$ и т. д. (если $y=0$) перейдет в слово $q_8 01^y 01^x$ или в слово $q_{17} 011^x 0$. Чтобы получить требуемое слово, далее присоединяем приказы

$$q_8 0 \to q_{18} R$$
$$q_{18} 1 \to q_{18} R$$
$$q_{18} 0 \to q_0 0 \quad 01^y q_0 01^x 0 \quad (y>0)$$
$$q_{17} 0 \to q_{19} R$$
$$q_{19} 1 \to q_0 0 \quad 01^y q_0 01^x 0 \quad (y=0).$$

г) Удвоение: $q_1 01^x 0^{x+3} \Longrightarrow q_0 01^x 01^x 00$.

Полагаем: $q_1 0 \to q_2 R$
$$q_2 1 \to q_2 R \quad 01^x q_2 0^{x+3}.$$

Полученное слово $01^x q_2 0^{x+3}$ представляем в виде $01^x q_2 00^0 00^0 0^{x+1}$ и далее пишем программу, для которой при $x-i>0$

$$01^{x-i} q_2 01^i 01^i 0 \Longrightarrow 01^{x-(i+1)} q_2 01^{i+1} 01^{i+1}.$$

Полагаем: $q_2 0 \to q_3 L \quad 01^{x-(i+1)} q_3 101^i 01^i 0$

$$q_3 1 \to q_4 0$$
$$q_4 0 \to q_5 R$$
$$q_5 0 \to q_6 1$$
$$q_6 1 \to q_6 R \quad 01^{x-(i+1)} 01^{i+1} q_6 01^i 0$$
$$q_6 0 \to q_7 R$$
$$q_7 1 \to q_7 R$$
$$q_7 0 \to q_8 1$$
$$q_8 1 \to q_8 L$$
$$q_8 0 \to q_9 L$$
$$q_9 1 \to q_9 L \quad 01^{x-(i+1)} q_9 01^{i+1} 01^{i+1}.$$

Теперь мы приказом

$$q_9 0 \to q_2 0 \quad 01^{x-(i+1)} q_2 01^{i+1} 01^{i+1}$$

зацикливаем программу и машина, пользуясь фиксированными выше приказами, начинает перерабатывать указанное слово. Если $x - (i+1) > 0$, то после цикла работы машины получится слово $01^{x-(i+2)} q_2 01^{i+2} 01^{i+2}$ и т. д. Заметим, что в течение каждого цикла машина вводит в игру новую ячейку ленты справа, беря ее из того запаса пустых ячеек 0^{x+3}, который нами был заранее написан в исходном слове. Если этого не было бы сделано, то машина пристраивала бы сама по одной новой пустой ячейке в каждом цикле работы.

Итак, имея начальное состояние $q_1 01^x 0^{x+3}$ и указанную выше программу, машина через x циклов работы придет в состояние $0q_2 01^x 01^x 0$. Приказ $q_2 0 \to q_3 L$ переведет машину в состояние $q_3 001^x 01^x 0$. Производя теперь перенос нуля **А**, затем правый сдвиг **Б**$^+$, еще раз перенос нуля **А** и, наконец, левый сдвиг **Б**$^-$, получим требуемое слово $q_0 01^x 01^x 00$.

Компонируя между собой машины **А**, **Б**$^+$, **Б**$^-$, **В**, **Г**, легко получить машины, выполняющие более сложные преобразования слов. Так, например, имеем:

Б$^+$**Б**$^+$) Двойной перенос:

$$\mathbf{Б}^+\mathbf{Б}^+\colon\ q_1 01^x 01^y 0 \Longrightarrow 01^x 01^y q_0 0.$$

Ц) Циклический сдвиг: **Ц** = **ВБ**$^-$**В**.

$$\mathbf{Ц}\colon\ 01^x 01^y q_1 01^z 0 \Longrightarrow 01^z q_0 01^x 01^y 0.$$

Гг) Копирование: **Гг** = **Б**$^+$**ГВБ**$^+$**ВГВБ**$^+$**ВБ**$^-$**Б**$^-$**ВБ**$^+$.

$$\mathbf{Гг}\colon\ q_1 01^x 01^y \Longrightarrow 01^x 01^y q_0 01^x 01^y.$$

Проверка того, что указанные композиции машин действительно выполняют заданные преобразования машинных слов, очевидна.

Как уже говорилось, для доказательства основной теоремы нам достаточно научиться строить машины, правильно вычисляющие простейшие функции и функции, возникающие из правильно вычислимых функций с помощью операций суперпозиции, примитивной рекурсии и минимизации. К этим построениям мы теперь и перейдем.

Лемма 2. *Функции* $\boldsymbol{o}(x) = 0$, $\boldsymbol{s}(x) = x + 1$, $\boldsymbol{d}(x) = x \dot{-} 1$ *правильно вычислимые.*

Программы машин, правильно вычисляющих указанные функции, пишутся очевидным образом:

$$\boldsymbol{o}(x)=0:\ q_1 01^x 0 \Longrightarrow q_0 00^{x+1}$$

$$\begin{array}{ll} q_1 0 \to q_2 R & \\ q_2 1 \to q_2 R & 01^x q_2 0 \\ q_2 0 \to q_3 L & \\ q_3 1 \to q_4 0 & 01^{x-1} q_4 00 \\ q_4 0 \to q_3 L & q_3 00^{x+1} \\ q_3 0 \to q_0 0 & q_0 00^{x+1} \end{array}$$

$$\boldsymbol{s}(x)=x+1:\quad q_1 01^x 0 \Longrightarrow q_0 01^{x+1}$$

$$\begin{array}{ll} q_1 0 \to q_2 R & \\ q_2 1 \to q_2 R & 01^x q_2 0 \\ q_2 0 \to q_3 1 & \\ q_3 1 \to q_3 L & \\ q_3 0 \to q_0 0 & q_0 01^{x+1} \end{array}$$

$$\boldsymbol{d}(x)=x \dot{-} 1:\ q_1 01^x 0 \Longrightarrow q_0 01^{x \dot{-} 1} 00$$

$$\begin{array}{ll} q_1 0 \to q_2 R & \\ q_2 1 \to q_2 R & 01^x q_2 0 \\ q_2 0 \to q_3 L & \\ q_3 1 \to q_4 0 & 01^{x-1} q_4 00 \\ q_4 0 \to q_5 L & \\ q_5 1 \to q_5 L & q_5 01^{x-1} 00 \\ q_5 0 \to q_0 0 & q_0 01^{x-1} 00 \\ q_3 0 \to q_0 0 & \text{для } x=0. \end{array}$$

Выше прослежен процесс переработки слова $q_1 01^x 0$ для $x \geqslant 1$. Легко убедиться, что результатом переработки указанного слова по изложенной программе для $x=0$ будет требуемое слово $q_0 00$.

Для того чтобы легче было следить за работой машин в оставшихся более сложных случаях, введем следующие обозначения. Пусть $\boldsymbol{T}$ — программа какой-то машины с алфавитами — внешним 0,1 и внутренним $q_0, q_1, \ldots, q_n$, причем q_0 — символ заключительного состояния. Пусть α — положительное натуральное число. Заменяя в записи

команд программы T символы $q_1, \ldots, q_n$ соответственно символами $q_\alpha, \ldots, q_{\alpha+n-1}$, а символ q_0 символом $q_{\alpha+n}$, получим программу машины с тем же внешним алфавитом 0,1 и внутренним алфавитом $q_\alpha, q_{\alpha+1}, \ldots, q_{\alpha+n}$, причем *заключительным символом теперь служит символ* $q_{\alpha+n}$. Эту новую программу мы будем обозначать через T_α или, подробнее, через $T_{\alpha\beta}$, где число $\beta=\alpha+n$ целиком определяется машиной T и числом α. Условимся также писать

$$\mathfrak{a}q_\alpha\mathfrak{b}\,(T)_\alpha\,\mathfrak{a}_1q_\beta\mathfrak{b}_1,$$

если машина T, начав работать в состоянии $\mathfrak{a}q_\alpha\mathfrak{b}$, переходит через некоторое время в *заключительное* состояние $\mathfrak{a}_1q_\beta\mathfrak{b}_10\ldots0$ при условии, что в процессе работы лента не надстраивалась слева. Вспоминая теперь определение композиции машин, непосредственно видим, что если для некоторых слов $\mathfrak{a}_0$, $\mathfrak{b}_0$, $\mathfrak{a}_1$, $\mathfrak{b}_1, \ldots, \mathfrak{a}_i$, $\mathfrak{b}_i$ в алфавите 0,1 и каких-то машин $U, V, \ldots, W$

$$\mathfrak{a}_0q_1\mathfrak{b}_0\,(U)_1\,\mathfrak{a}_1q_\beta\mathfrak{b}_1\,(V)_\beta\,\mathfrak{a}_2q_\gamma\mathfrak{b}_2\ldots(W)_\delta\,\mathfrak{a}_iq_\varepsilon\mathfrak{b}_i,$$

то

$$\mathfrak{a}_0q_1\mathfrak{b}_0\,((UV)\ldots W)\,\mathfrak{a}_iq_0\mathfrak{b}_i.$$

Докажем теперь оставшиеся нужные нам леммы.

Лемма 3. *Простейшие функции* $I_t^s(x_1, \ldots, x_s)=x_t$ ($s\geqslant t$; $s, t=1, 2, \ldots$) *правильно вычислимы.*

Укажем, например, программу для I_2^2. Символы **А**, **Б**, **В**, **Г**, **(0)**, ... обозначают программы рассмотренных выше машин. Очевидно,

$$q_101^x01^y\,(\mathbf{Б}^+)_1\,01^xq_\beta01^y\,(\mathbf{В})_\beta\,01^yq_\gamma01^x\,(\mathbf{0})_\gamma\,01^yq_\delta0\ldots$$
$$\ldots0\,(\mathbf{Б}^-)_\delta q_\varepsilon01^y.$$

Таким образом, искомой машиной, вычисляющей правильно функцию I_2^2, является композиция $\mathbf{Б}^+*\mathbf{В}*(\mathbf{0})*\mathbf{Б}^-$.

Лемма 4. *Если частичные функции* $g(x_1, \ldots, x_s)$, $h_1(x_1, \ldots, x_t), \ldots, h_s(x_1, \ldots, x_t)$ *правильно вычислимы, то суперпозиция*

$$f(x_1, \ldots, x_t)=g(h_1(x_1, \ldots, x_t), \ldots, h_s(x_1, \ldots, x_t))$$

— также правильно вычислимая частичная функция.

Пусть (g), $(h_1), \ldots, (h_s)$ — программы машин, правильно вычисляющих указанные функции. Если функция g

одноместная, то из

$$q_1 01^{x_1} \ldots 01^{x_t} (h_1)_1\, q_\beta 01^{h_1(x_1, \ldots, x_t)} (g)_\beta\, q_\gamma 01^{g(h_1)}$$

видно, что машина $(h_1)*(g)$ правильно вычисляет нужную нам функцию $g\,(h_1)$.

Чтобы не писать излишне длинных выражений, рассмотрим еще лишь случай, когда заданные частичные функции g, h_1, h_2 двуместные. Пользуясь введенными выше машинами сдвига, транспозиции, удвоения и нулевой функции, получим

$$\begin{array}{rl} q_1 01^x 01^y\, (\mathbf{Гг})_1 & 01^x 01^y q_\alpha 01^x 01^y \\ (h_1)_\alpha & 01^x 01^y q_\beta 01^{h_1(x,\, y)} \\ (\mathbf{Ц})_\beta & 01^{h_1(x,\, y)} q_\gamma 01^x 01^y \\ (h_2)_\gamma & 01^{h_1(x,\, y)} q_\delta 01^{h_2(x,\, y)} \\ (\mathbf{Б}^-)_\delta & q_\varepsilon 01^{h_1(x,\, y)} 01^{h_2(x,\, y)} \\ (g)_\varepsilon & q_\varkappa 01^{g(h_1,\, h_2)} \\ q_\varkappa 0 \to q_0 0 & q_0 01^{g(h_1,\, h_2)}. \end{array}$$

Лемма 5. *Функция f, возникающая примитивной рекурсией из правильно вычислимых функций g, h, сама правильно вычислима.*

Для наших целей достаточно будет рассмотреть частный случай, когда функция f связана с функциями g, h зависимостью

$$f(x,\, 0) = g(x),$$
$$f(x,\, i+1) = h(f(x,\, i),\, x).$$

Как и выше, через (g), (h) обозначим программы машин, правильно вычисляющих функции g и h. Пусть x, y — произвольные натуральные числа. Имеем:

$$\begin{array}{rl} & q_1 01^x 01^y 0 \\ (\mathbf{Б}^+\mathbf{ВГВБ}^+)_1 & 01^x 01^y q_\alpha 01^x 0 \\ (g)_\alpha & 01^x 01^y q_\beta 01^{f(x,\, 0)} \\ q_\beta 0 \to q_{\beta+1} L & \\ q_{\beta+1} 1 \to q_{\beta+2} 0 & \end{array}$$

$$\begin{array}{rl} (\mathbf{A})_{\beta+2} & 01^x01^{y-1}q_\gamma 01^{f(x,\,0)} \\ (\mathbf{Б^-ВБ^+ВГ})_\gamma & 01^{y-1}01^{f(x,\,0)}q_\delta 01^x01^x \\ (\mathbf{ВБ^-ВБ^+})_\delta & 01^x01^{y-1}q_\varepsilon 01^{f(x,\,0)}01^x \\ (h)_\varepsilon & 01^x01^{y-1}q_\varkappa 01^{f(x,\,1)} \end{array}$$

$$q_\varkappa 0 \to q_\beta 0.$$

Команды, идущие после первого появления состояния q_β, преобразуют машинное слово

$$01^x01^{y-i}q_\beta 01^{f(x,\,i)}$$

в слово

$$01^x01^{y-(i+1)}q_\varkappa 01^{f(x,\,i+1)}$$

при условии, что $y > i$. Команда $q_\varkappa 0 \to q_\beta 0$ зацикливает программу и последующими преобразованиями параметр $y - i$ будет понижаться до тех пор, пока не получится слово

$$01^x0q_\beta 01^{f(x,\,y)},$$

которое в силу команды $q_\beta 0 \to q_{\beta+1}L$ перейдет в слово

$$01^xq_{\beta+1}001^{f(x,\,y)}.$$

Дополнительными командами

$$q_{\beta+1}0 \to q_{\varkappa+1}0,\ (\mathbf{A})_{\varkappa+1},\ (\mathbf{В}\ (\mathbf{0})\ \mathbf{В})_\lambda,\ (\mathbf{Б^-})_\mu,\ (\mathbf{A})_\nu$$

получим требуемое состояние $q_0 01^{f(x,\,y)}$.

Отметим несколько следствий, непосредственно вытекающих из доказанных лемм. Прежде всего, соотношения

$$x+0=x, \quad x+(i+1)=(x+i)+1$$

показывают, что функция $x+y$ получается из функций $g(x)=x$ и $h(x, y)=x+1$ посредством примитивной рекурсии рассмотренного в лемме 5 вида. Функции x и $s\ (I_1^2)$ согласно леммам 2, 3 правильно вычислимы. Поэтому и функция $x+y$ правильно вычислима.

Аналогичным образом убеждаемся, что функции $x \dot{-} y$ и xy возникают примитивной рекурсией указанного в лемме 5 вида из функций x, $h(x, y) = x \dot{-} 1$ и соответственно $o(x)$, $y+x$. Так как последние функции правильно вычис-

лимы, то функции $x \dot{-} y$ и xy также правильно вычислимы. В силу леммы 4 вместе с функциями x, $x_1 \cdot x_2$ правильно вычислимой будет и функция x^2.

Лемма 6. *Если частичная функция $g(x, y)$ правильно вычислима, то функция*

$$f(x) = \mu_y (g(x, y) = 0)$$

также правильно вычислима.

Действительно, для любого заданного натурального x

$$\begin{array}{ll} & q_1 01^x 0 \\ (\text{Гг})_1 & 01^x 0 q_\alpha 01^x 0 \\ (g)_\alpha & 01^x 01^0 q_\beta 01^{g(x,\,0)}. \end{array}$$

Теперь мы, как и выше, подбираем команды, которые при условии $g(x, i) > 0$ преобразовывали бы слово $01^x 01^i q_\beta 01^{g(x,\,i)}$ в слово $01^x 01^{i+1} q_\beta 01^{g(x,\,i+1)}$:

$$\begin{array}{rl} q_\beta 0 \to q_{\beta+1} R & \\ q_{\beta+1} 1 \to q_{\beta+2} 0 & \\ q_{\beta+2} 0 \to q_{\beta+3} L & \\ q_{\beta+3} 0 \to q_{\beta+4} 1 & \\ q_{\beta+4} 1 \to q_{\beta+5} R & 01^x 01^1 q_{\beta+5} 01^{g(x,\,0)-1} \\ (\mathbf{0})_{\beta+5} & 01^x 01^1 q_\gamma 0 \\ (\text{Б}^-\text{Б}^-)_\gamma & q_\delta 01^x 01^1 \\ (\text{Гг})_\delta & 01^x 01^1 q_\varepsilon 01^x 01^1 \\ (g)_\varepsilon & 01^x 01^1 q_\varkappa 01^{g(x,\,1)} \\ g_\varkappa 0 \to g_\beta 0 & 01^x 01^1 q_\beta 01^{g(x,\,1)}. \end{array}$$

Последняя команда зацикливает программу и машина от состояния $01^x 01^1 q_\beta 01^{g(x,\,1)}$ преобразуется в состояние $01^x 01^2 q_\beta 01^{g(x,\,2)}$, затем в состояние $01^x 01^3 q_\beta 01^{g(x,\,3)}$ и т. д. Допустим, что по истечении некоторого времени машина перешла в состояние $01^x 01^i q_\beta 01^{g(x,\,i)}$ и $g(x, i) = 0$. Написанная выше команда $q_\beta 0 \to q_{\beta+1} R$ переведет машину в состояние $01^x 01^i 0 q_{\beta+1} 0$. Так как в рассматриваемом случае $f(x) = i$, то нам остается дописать лишь команды, под

воздействием которых машина из указанного состояния перейдет в состояние $q_0 01^i$. Полагаем

$$\begin{array}{rl} q_{\beta+1}0 \to q_{\varkappa+1}0 & 01^x01^i0q_{\varkappa+1}0 \\ (\text{Б}^-\text{Б}^-)_{\varkappa+1} & 01^x q_\lambda 01^i \\ (\text{В}\ (0)\ \text{Б}^-)_\lambda & q_\mu 01^i \\ q_\mu 0 \to q_0 0 & q_0 01^{f(x)}. \end{array}$$

Если в процессе вычисления значений $g(x, i)$ $(i = 0, 1, 2, \ldots)$ встретится либо неопределенное значение, либо все значения будут определенные, но отличные от 0, то машина будет работать вечно, не приходя никогда во внутреннее состояние q_0. Но в этих случаях функция $f(x)$ имеет неопределенное значение и потому во всех случаях указанная программа будет вычислять $f(x)$. Лемма доказана.

Выше была установлена правильная вычислимость функций $\boldsymbol{o}(x)$, $x + 1$, x^2, $x \dot{-} y$, $x + y$, I_m^n, $\overline{\text{sg}}$. В силу леммы 4 вместе с этими функциями правильно вычислимыми будут и все функции, получающиеся из заданных операциями подстановки. В частности, правильно вычислимыми будут функции $\overline{\text{sg}}\,((y + 1)^2 \dot{-} x)$ и $\overline{\text{sg}}\,(2(y + 1) \dot{-} x)$. Но

$$[x/2] = \mu_y\,(\overline{\text{sg}}\,(2(y+1) \dot{-} x) = 0),$$

$$[\sqrt{x}] = \mu_y\,(\overline{\text{sg}}\,((y+1)^2 \dot{-} x) = 0),$$

поэтому в силу леммы 6 функции $[x/2]$, $[\sqrt{x}]$ также правильно вычислимы.

В п. 3.3 были введены нумерационные функции Кантора $\boldsymbol{c}(x, y)$, $\boldsymbol{l}(x)$, $\boldsymbol{r}(x)$. Из явных выражений для этих функций видно, что они выражаются через x, y при помощи $+$, $\dot{-}$, x^2, $[\sqrt{x}]$, $[x/2]$. Поэтому функции $\boldsymbol{c}$, $\boldsymbol{l}$, $\boldsymbol{r}$ правильно вычислимы.

Теперь для завершения доказательства теоремы о синтезе машин Тьюринга нам остается лишь сослаться на следствие теоремы 2 из п. 3.4, согласно которому любая частично рекурсивная функция может быть получена из функций $\boldsymbol{o}$, $x+1$, I_n^m, $\boldsymbol{c}$, $\boldsymbol{l}$, $\boldsymbol{r}$ конечным числом операций подстановки, рекурсий из леммы 5 и минимизаций из леммы 6.

12.4. Теоремы о графике и существовании универсальных частично рекурсивных функций. В пп. 6.1 и 6.2 были доказаны теоремы о рекурсивной перечислимости графика произвольной частично рекурсивной функции и о существовании универсальной частично рекурсивной функции. Однако доказательства этих фундаментальных фактов были довольно длинными и опирающимися на ряд лемм. Мы теперь хотим дать новый, краткий и прозрачный вывод указанных теорем, опирающийся лишь на развитую выше теорию машин Тьюринга.

Теорема 1. *Область определенности произвольной частично рекурсивной числовой функции* $f(x_1, \ldots, x_n)$ *является рекурсивно перечислимым множеством.*

Пусть $q_0, q_1, \ldots, q_m$ — внутренний алфавит машины Тьюринга $\mathfrak{T}$, правильно вычисляющей функцию f. Это означает, что

$$\mathfrak{T}\colon\ q_1 01^{x_1} \ldots 01^{x_n} \Longrightarrow q_0 01^{f(x_1, \ldots, x_n)} 0 \ldots 0$$

для тех $x_1, \ldots, x_n$, для которых f определена, и что $\mathfrak{T}$ работает вечно для остальных $x_1, \ldots, x_n$. В соответствии с п. 12.2 обозначим через $(q_1\mathfrak{a})^{(t)}$ машинное слово, получающееся из слова $q_1\mathfrak{a}$ после t тактов работы $\mathfrak{T}$. Согласно п. 12.2 словарная функция

$$P(\mathfrak{x}_1, \ldots, \mathfrak{x}_n, \mathfrak{y}) = \left(q_1 01^{(\text{длина } \mathfrak{x}_1)} \ldots 01^{(\text{длина } \mathfrak{x}_n)}\right)^{(\text{длина } \mathfrak{y})}$$

примитивно рекурсивна. Нам надо найти такой момент t, при котором буква q_0 войдет в слово P, то есть при котором

$$E(q_0, P(1^{x_1}, \ldots, 1^{x_n}, 1^t)) = \Lambda, \qquad (1)$$

где E — словарная функция, определенная в п. 11.2. Так как словарные функции E и P примитивно рекурсивны, то примитивно рекурсивной будет и числовая функция

$$g(x_1, \ldots, x_n, t) = cE(q_0, P(1^{x_1}, \ldots, 1^{x_n}, 1^t)).$$

По условию $f(x_1, \ldots, x_n)$ определено для тех и только тех $x_1, \ldots, x_n$, для которых существует решение t уравнения (1), то есть для которых разрешимо уравнение

$$g(x_1, \ldots, x_n, t) = 0,$$

а это означает согласно п. 4.4, что область определенности функции f рекурсивно перечислима.

Следствие. *График произвольной частично рекурсивной функции $f(x_1, \ldots, x_n)$ рекурсивно перечислим.*

Рассмотрим частичную функцию

$$(y \dot{-} f(x_1, \ldots, x_n)) + (f(x_1, \ldots, x_n) \dot{-} y).$$

Она частично рекурсивна и ее область определенности совпадает с графиком функции f. Поэтому график f рекурсивно перечислим.

Теорема 2. *Существует частично рекурсивная функция двух переменных $T(x_0, x_1)$, обладающая следующим свойством: последовательность одноместных частично рекурсивных функций*

$$T(0, x),\ T(1, x),\ \ldots,\ T(n, x),\ \ldots$$

содержит каждую одноместную частично рекурсивную функцию.

Функции $T(n, x)$, обладающие указанным свойством, обычно называют *универсальными частично рекурсивными* функциями двух переменных. Переменные x_0, x_1 в теореме 2 играют разные роли. Поэтому первую из них часто называют параметром. Конечно, можно сформулировать аналогичную теорему и относительно существования универсальных частично рекурсивных функций многих переменных. Однако все эти функции легко получаются (см. п. 6.2) из универсальной функции двух переменных.

Идея построения функции $T(n, x)$ следующая. Каждая одноместная частично рекурсивная функция $f(x)$ правильно вычислима на подходящей машине Тьюринга $\mathfrak{T}$ с внешним алфавитом 0,1 и внутренним алфавитом $q_0, q_1, \ldots, q_n$ (n зависит от f). Машина $\mathfrak{T}$ вполне определяется своей таблицей переходов

$$q_i 0 \to q_{\alpha_{i0}} v_{i0}$$

$$q_i 1 \to q_{\alpha_{i1}} v_{i1} \qquad (v_{ij} = 0,\ 1,\ L,\ R; \quad i = 1, \ldots, n).$$

Всевозможные таблицы этого вида можно перенумеровать и постараться определить такую частично рекурсивную функцию двух переменных $T(n, x)$, что если частичная функция $f(x)$ правильно вычисляется на машине с номером n, то $f(x) = T(n, x)$. Так как каждая одноместная частично рекурсивная функция правильно вычисляется

на подходящей машине Тьюринга, то при соблюдении указанного условия функция $T(n, x)$ будет заведомо универсальной.

Рассмотрим алфавит $\boldsymbol{A} = \{1, 0, q, L, R, e\}$. Условимся в алфавите $\boldsymbol{A}$ внутреннее состояние q_i изображать словом q^{i+1} $(i = 0, 1, \ldots, n)$. В частности, машинное слово $q_1 01^x$ в алфавите $\boldsymbol{A}$ будет изображаться словом $q^2 01^x$, машинное слово $q_0 01^x 0 \ldots 0$ будет изображаться словом $q01^x0 \ldots 0$ и т. д.

Шифром машины $\mathfrak{T}$ в алфавите $\boldsymbol{A}$ условимся называть слово

$$\mathfrak{a} = eq^2 0 q^{\alpha_{10}} v_{10} e q^2 1 q^{\alpha_{11}} v_{11} \ldots e q^{n+1} 0 q^{\alpha_{n0}} v_{n0} e q^{n+1} 1 q^{\alpha_{n1}} v_{n1},$$

где значения α_{ij}, v_{ij} берутся из упомянутой выше таблицы переходов машины $\mathfrak{T}$. Ясно, конечно, что, зная шифр машины, мы тем самым знаем и таблицу переходов. Номер шифра часто называют номером самой машины $\mathfrak{T}$. Однако далее мы не будем пользоваться номерами машин Тьюринга, а будем рассматривать лишь шифры машин.

Допустим теперь, что нам удалось построить определенную в алфавите $\boldsymbol{A}$ трехместную примитивно рекурсивную словарную функцию $M(\mathfrak{a}, \mathfrak{x}, \mathfrak{y})$, удовлетворяющую требованию

а) *если* $\mathfrak{a}$ — *шифр некоторой машины Тьюринга* $\mathfrak{T}$, *то*

$$M(\mathfrak{a}, q^2 01^x, 1^y) = (q^2 01^x)^{(y)} \qquad (x, y = 0, 1, \ldots).$$

Тогда требуемая универсальная частично рекурсивная функция $T(n, x)$ может быть определена следующим очевидным способом. Полагаем

$$P(\mathfrak{a}, \mathfrak{x}) = M(\mathfrak{a}, \mathfrak{x}, \mu 1^y (\boldsymbol{E}^*(q^2, M(\mathfrak{a}, \mathfrak{x}, 1^y)) = \Lambda)), \qquad (2)$$

где

$$\boldsymbol{E}^*(\mathfrak{a}, \mathfrak{b}) = \boldsymbol{W}_1(1, \Lambda; \boldsymbol{E}(\mathfrak{a}, \mathfrak{b})) = \begin{cases} 1, & \text{если } \mathfrak{a} \in \mathfrak{b}, \\ \Lambda, & \text{если } \mathfrak{a} \notin \mathfrak{b}. \end{cases}$$

Иными словами, если $\mathfrak{a}$ — шифр машины $\mathfrak{T}$ и

$$1^z = \mu 1^y (\boldsymbol{E}^*(q^2, M(\mathfrak{a}, q^2 01^x, 1^y)) = \Lambda),$$

то машина $\mathfrak{T}$, начав работать в конфигурации $q^2 01^x$, через z тактов работы придет в заключительное состояние, так как машинное слово $(q^2 01^x)^{(z)}$ не будет содержать подслов

вида q^i, $i \geqslant 2$, характеризующих активное внутреннее состояние $\mathfrak{T}$. Сама заключительная конфигурация будет согласно а) характеризоваться словом $M(\mathfrak{a}, q^2 01^x, 1^z)$. В частности, если машина $\mathfrak{T}$ правильно вычисляет функцию $f(x)$, то для каждого натурального x имеем

$$P(\mathfrak{a}, q^2 01^x) = q01^{f(x)} 0 \ldots 0. \qquad (3)$$

Чтобы избавиться в формуле (3) от стоящих в конце букв 0, введем особую функцию $G_{\mathrm{I}}(\mathfrak{x})$, полагая по определению: $G_{\mathrm{I}}(\mathfrak{x})$ = первое максимальное подслово вида 1^{i+1} в $\mathfrak{x}$, $G_{\mathrm{I}}(\mathfrak{x}) = \Lambda$, если $\mathfrak{x}$ не содержит подслов вида 1^{i+1}.

Примитивная рекурсивность функций $G_{\mathrm{I}}(\mathfrak{x})$ будет доказана ниже.

Из формулы (3) и определения функции G_{I} получаем

$$G_{\mathrm{I}}(P(\mathfrak{a}, q^2 01^x)) = 1^{f(x)}. \qquad (4)$$

Функции M, E^* примитивно рекурсивны (п. 11.2). Формула (2) показывает, что функция $P(\mathfrak{a}, \mathfrak{x})$ частично рекурсивна. Так как функция G_{I} примитивно рекурсивна, то функция

$$H(\mathfrak{a}, \mathfrak{x}) = G_{\mathrm{I}}(P(\mathfrak{a}, \mathfrak{x}))$$

также частично рекурсивна. Вместе с нею будет частично рекурсивной и ее представляющая функция

$$h(a, x) = c(H(c^{-1}a, c^{-1}x)),$$

где через $c^{-1}n$ обозначается слово, имеющее словарный номер n (п. 11.1). Введем еще числовые функции

$$g(x) = c(q^2 01^x),$$
$$k(x) = \mu_y(c(1^y) = x).$$

Очевидно, первая из этих функций примитивно рекурсивна, а вторая частично рекурсивна. Поэтому функция

$$T(a, x) = k(h(a, g(x)))$$

является частично рекурсивной. *Она и является искомой универсальной частично рекурсивной функцией.* Действительно, пусть $f(x)$ — какая-то частично рекурсивная одноместная функция. Согласно п. 12.3 $f(x)$ правильно вычислима на подходящей машине Тьюринга $\mathfrak{T}$. Пусть $\mathfrak{a}$ — шифр

этой машины в алфавите A и $a = c(\mathfrak{a})$ — номер этого шифра. Из формулы (4) получаем

$$H(\mathfrak{a}, q^2 01^x) = 1^{f(x)}$$

и, следовательно,

$$h(a, g(x)) = c(1^{f(x)}),$$
$$k(h(a, g(x))) = f(x),$$

то есть

$$T(a, x) = f(x),$$

а это и означает, что $T(n, x)$ — частично рекурсивная универсальная функция.

Нам остается показать, что примитивно рекурсивная словарная функция $M(\mathfrak{a}, \mathfrak{x}, \mathfrak{y})$, обладающая свойством а), существует. Но функция M, определяемая словарной рекурсией

$$M(\mathfrak{a}, \mathfrak{x}, \Lambda) = \mathfrak{x}, \quad M(\mathfrak{a}, \mathfrak{x}, \mathfrak{y}u) = \Phi(\mathfrak{a}, M(\mathfrak{a}, \mathfrak{x}, \mathfrak{y})) \qquad (u \in A),$$

заведомо обладает свойством а), если вспомогательная функция $\Phi(\mathfrak{a}, \mathfrak{x})$ примитивно рекурсивна и удовлетворяет следующему требованию:

б) если $\mathfrak{a}$ — шифр некоторой машины $\mathfrak{T}$ и $\mathfrak{x}$ — произвольное машинное слово, то $\Phi(\mathfrak{a}, \mathfrak{x})$ есть машинное слово, описывающее состояние машины $\mathfrak{T}$, в которое она переходит из состояния $\mathfrak{x}$ после одного такта работы.

В свою очередь, функцию Φ мы будем искать в виде

$$\Phi(\mathfrak{a}, \mathfrak{x}) = \mathrm{Com}(\Psi(\mathfrak{a}, \mathfrak{x}), \mathfrak{x}), \tag{5}$$

где Com, Ψ — подходящие примитивно рекурсивные функции. Ясно, что функция Φ, определенная формулой (5), заведомо удовлетворяет требованию б), если функции Ψ и Com удовлетворяют соответственно условиям

в) если $\mathfrak{a}$ — шифр машины Тьюринга $\mathfrak{T}$ и $\mathfrak{x}$ — машинное слово $\mathfrak{x} = \mathfrak{p}q^{i+1}u\mathfrak{q}$ $(u = 0, 1)$, то

$$\Psi(\mathfrak{a}, \mathfrak{x}) = q^{j+1}v \qquad (v = 0, 1, L, R),$$

где $q^{j+1}v$ — правая часть приказа $q^{i+1}u \to q^{j+1}v$, выполняемого машиной $\mathfrak{T}$;

г) если $\mathfrak{x}$ — машинное слово $\mathfrak{x} = \mathfrak{p}q^{i+1}u\mathfrak{q}$, а $\mathfrak{b}$ равно какому-нибудь слову вида $q^{j+1}v$ $(v = 0, 1, L, R)$, то

Com $(\mathfrak{b}, \mathfrak{x})$ равно машинному слову, получающемуся из $\mathfrak{x}$ командой $q^{i+1}u \to \mathfrak{b}$.

Строим сначала функцию Ψ. Функция St (x) = *начальной букве x* удовлетворяет словарной рекурсии: St $(\Lambda) = \Lambda$ и

$$\mathrm{St}(\mathfrak{x}u) = \begin{cases} \mathrm{St}(\mathfrak{x}), & \text{если } \mathfrak{x} \neq \Lambda, \\ u, & \text{если } \mathfrak{x} = \Lambda, \end{cases} = \boldsymbol{W}_1(u, \mathrm{St}\mathfrak{x}; \mathfrak{x})$$

и потому примитивно рекурсивна. Функция Rest $(\mathfrak{a}, \mathfrak{x})$ = = части слова $\mathfrak{x}$, лежащей за первым вхождением $\mathfrak{a}$ в $\mathfrak{x}$, если $\mathfrak{a} \in \mathfrak{x}$, и Rest $(\mathfrak{a}, \mathfrak{x}) = \Lambda$, если $\mathfrak{a} \notin \mathfrak{x}$, удовлетворяет словарной рекурсии: Rest $(\mathfrak{a}, \Lambda) = \Lambda$ и

$$\mathrm{Rest}(\mathfrak{a}, \mathfrak{x}u) = \begin{cases} \mathrm{Rest}(\mathfrak{a}, \mathfrak{x})\, u, & \textit{если } \mathfrak{a} \in \mathfrak{x}, \\ \Lambda, & \textit{если } \mathfrak{a} \notin \mathfrak{x}, \end{cases} =$$
$$= \boldsymbol{W}_1(\mathrm{Rest}(\mathfrak{a}, \mathfrak{x})\, u, \Lambda; \boldsymbol{E}(\mathfrak{a}, \mathfrak{x})).$$

Поэтому функция Rest примитивно рекурсивна. Наконец, функции $G_u(\mathfrak{x})$ $(u \in \boldsymbol{A})$, определенные схемой: $G_u(\mathfrak{x}) =$ = *первому максимальному подслову вида* u^{i+1}, *входящему в* $\mathfrak{x}$, *если* $\mathfrak{x}$ *содержит подслова этого вида, и* $G_u(\mathfrak{x}) = \Lambda$, *если* $\mathfrak{x}$ *не содержит подслов указанного вида*, удовлетворяют словарной рекурсии: $G_u(\Lambda) = \Lambda$ и

$$G_u(\mathfrak{x}v) = G_u(\mathfrak{x}) \qquad (v \in \boldsymbol{A},\ v \neq u),$$
$$G_u(\mathfrak{x}u) = \begin{cases} G_u(\mathfrak{x}), & \textit{если } \mathrm{Rest}(G_u(\mathfrak{x}), \mathfrak{x}) \neq \Lambda, \\ G_u(\mathfrak{x})\, u, & \textit{если } \mathrm{Rest}(G_u(\mathfrak{x}), \mathfrak{x}) = \Lambda. \end{cases}$$

Поэтому функции $G_u(\mathfrak{x})$ примитивно рекурсивны.

Из функций $G_u(\mathfrak{x})$ выше была использована функция $G_{\mathrm{I}}(\mathfrak{x})$. Далее нам будет нужна лишь функция $G_q(\mathfrak{x}) =$ $= G(\mathfrak{x})$. Исходя из этой функции и функций St, Sb (п. 11.2), легко уже построить и функцию Ψ, обладающую свойством в). В самом деле, пусть $\mathfrak{x} = \mathfrak{p}q^{i+1}u\mathfrak{q}$ $(u = 0, 1)$ — машинное слово и

$$\mathfrak{a} = eq^2 0 q^{\alpha_{10}} v_{10} e q^2 1 q^{\alpha_{11}} v_{11} \ldots e q^{n+1} 0 q^{\alpha_{n0}} v_{n0} e q^{n+1} 1 q^{\alpha_{n1}} v_{n1}$$

— шифр некоторой машины $\mathfrak{T}$. Тогда

$$G(\mathfrak{x}) = q^{i+1},$$
$$\mathrm{St}(\mathrm{Rest}(G(\mathfrak{x}), \mathfrak{x})) = u.$$

Вводя примитивно рекурсивную функцию

$$\mathrm{Dir}\,\mathfrak{x} = G(\mathfrak{x})\,\mathrm{St}\,(\mathrm{Rest}\,(G(\mathfrak{x}),\ \mathfrak{x})),$$

будем иметь

$$\mathrm{Dir}\,\mathfrak{x} = q^{i+1}u,$$

$$\mathrm{Rest}\,(e\,\mathrm{Dir}\,\mathfrak{x},\ \mathfrak{a}) = q^{\alpha_{iu}}v_{iu}\ \ldots\ eq^{n+1}1q^{\alpha_{n1}}v_{n1},$$

$$\mathrm{Dir}\,(\mathrm{Rest}\,(e\,\mathrm{Dir}\,\mathfrak{x},\ \mathfrak{a})) = q^{\alpha_{iu}}v_{iu}.$$

Поэтому примитивно рекурсивная функция

$$\Psi(\mathfrak{a},\ \mathfrak{x}) = \mathrm{Dir}\,(\mathrm{Rest}\,(e\,\mathrm{Dir}\,\mathfrak{x},\ \mathfrak{a}))$$

заведомо удовлетворяет условию в).

Аналогичным образом строится и функция Com. Заметим сначала, что равенство $\mathrm{St}\,(\mathfrak{b}^{\sim}) = u$ ($\mathfrak{b}^{\sim}$ есть обращение слова $\mathfrak{b}$ (см. п. 11.2)) означает, что u есть *последняя* буква слова $\mathfrak{b}$. Условие г) мы теперь запишем на формульном языке в виде следующей схемы:

$$\mathrm{Com}\,(\mathfrak{b},\ \mathfrak{x}) =$$

$$= \begin{cases} \mathrm{Sb}\,(\mathfrak{x};\ \mathrm{Dir}\,\mathfrak{x},\ \mathfrak{b}), & \text{если } \mathrm{St}\,(\mathfrak{b}^{\sim}) = 0,\ 1, \\ \mathrm{Sb}\,(\mathfrak{x};\ \mathrm{Dir}\,\mathfrak{x},\ uG(\mathfrak{b})), & \text{если } \mathrm{St}\,(\mathfrak{b}^{\sim}) = R,\ B\mathfrak{x}=u,\ C\mathfrak{x}\neq\Lambda, \\ & \qquad (u \in A), \\ \mathrm{Sb}\,(\mathfrak{x};\ \mathrm{Dir}\,\mathfrak{x},\ uG(\mathfrak{b})\,0), & \text{если } \mathrm{St}\,(\mathfrak{b}^{\sim}) = R,\ B\mathfrak{x}=u,\ C\mathfrak{x}=\Lambda, \\ \mathrm{Sb}\,(\mathfrak{x};\ (\mathrm{Dir}\,\mathfrak{x}^{\sim}),\ G(\mathfrak{b})\,u), & \text{если } \mathrm{St}\,(\mathfrak{b}^{\sim}) = L,\ B\mathfrak{x}^{\sim}=u,\ D\mathfrak{x}\neq\Lambda, \\ \mathrm{Sb}\,(\mathfrak{x};\ G(\mathfrak{x}),\ G(\mathfrak{b})\,0), & \text{если } \mathrm{St}\,(\mathfrak{b}^{\sim}) = L,\ D\mathfrak{x} = \Lambda, \\ \Lambda & \text{в остальных случаях}, \end{cases}$$

где для краткости положено

$$B\mathfrak{x} = \mathrm{St}\,\mathrm{Rest}\,(G\mathfrak{x};\ \mathfrak{x}),\ C\mathfrak{x} = \mathrm{Rest}\,(\mathrm{Dir}\,\mathfrak{x},\ \mathfrak{x}),$$

$$D\mathfrak{x} = \mathrm{Rest}\,((\mathrm{Dir}\,\mathfrak{x})^{\sim},\ \mathfrak{x}^{\sim}).$$

Эта схема определяет значение функции $\mathrm{Com}\,(\mathfrak{b},\ \mathfrak{x})$ для любых слов $\mathfrak{b}$, $\mathfrak{x}$ в алфавите $\boldsymbol{A}$. Она подобрана так, что Com, как легко видеть, заведомо удовлетворяет требованию г). Остается лишь убедиться, что функция Com примитивно рекурсивна. Для этого мы преобразуем схему, заменяя указанные в ней равенства и неравенства стандартными равенствами пустому слову при помощи следующих

соотношений:

$$\mathfrak{a} = \mathfrak{b} \Leftrightarrow \boldsymbol{E}(\mathfrak{a}, \mathfrak{b})\,\boldsymbol{E}(\mathfrak{b}, \mathfrak{a}) = \Lambda,$$
$$\mathfrak{a} \neq \Lambda \Leftrightarrow \boldsymbol{W}_1(1, \Lambda; \mathfrak{a}) = \Lambda,$$
$$\mathfrak{a} = \Lambda \text{ и } \mathfrak{b} = \Lambda \Leftrightarrow \mathfrak{a}\mathfrak{b} = \Lambda.$$

После этого определяющая функцию Com схема примет вид

$$\mathrm{Com}(\mathfrak{b}, \mathfrak{x}) = \begin{cases} \mathfrak{y}_1, & \text{если } \mathfrak{a}_1 = \Lambda, \\ \dots\dots\dots & \\ \mathfrak{y}_p, & \text{если } \mathfrak{a}_p = \Lambda, \\ \Lambda & \text{в остальных случаях.} \end{cases} \qquad (6)$$

Схема (6) равносильна равенству

$$\mathrm{Com}(\mathfrak{b}, \mathfrak{x}) = W_p(\mathfrak{y}_1, \dots, \mathfrak{y}_p, \Lambda;\ \mathfrak{a}_1, \dots, \mathfrak{a}_p).$$

Так как функции $\mathfrak{y}_1, \dots, \mathfrak{y}_p, \mathfrak{a}_1, \dots, \mathfrak{a}_p$ примитивно рекурсивны, то из этого равенства следует, что функция Com также примитивно рекурсивна.

Итак, примитивно рекурсивная словарная функция $M(\mathfrak{a}, \mathfrak{x}, \mathfrak{y})$, обладающая свойством а), построена.

12.5. Универсальные машины. В п. 8.1 множество натуральных чисел $\boldsymbol{U}$ было названо $\boldsymbol{m}$-универсальным, если $\boldsymbol{U}$ рекурсивно перечислимо и для каждого рекурсивно перечислимого множества $\boldsymbol{S}$ существует такая общерекурсивная функция $g(x)$, что

$$x \in \boldsymbol{S} \Leftrightarrow g(x) \in \boldsymbol{U}.$$

Множество $\mathfrak{B}$ слов в произвольном алфавите $\boldsymbol{B}$ называется $\boldsymbol{m}$-универсальным, если совокупность номеров слов из $\mathfrak{B}$ $\boldsymbol{m}$-универсальна.

Если вместо алфавита $\boldsymbol{B}$ мы будем рассматривать более широкий алфавит $\boldsymbol{A}$, то номера слов из $\mathfrak{B}$ в алфавите $\boldsymbol{A}$ будут иными. Однако легко убедиться (см. п. 11.2), что если совокупность номеров слов из $\mathfrak{B}$, вычисленная в алфавите $\boldsymbol{B}$, является $\boldsymbol{m}$-универсальной, то совокупность номеров слов из $\mathfrak{B}$, вычисленных в алфавите $\boldsymbol{A}$, будет также $\boldsymbol{m}$-универсальной. В частности, если алфавит состоит лишь из одного символа a, то словарный номер слова a^x совпадает с x. Поэтому совокупность слов $\{a^{x_0}, a^{x_1}, \dots\}$ (в произвольном алфавите, содержащем a) тогда и только тогда

$\mathfrak{m}$-универсальна, когда $\mathfrak{m}$-универсальна совокупность чисел $\{x_0, x_1, \ldots\}$.

После этих предварительных замечаний рассмотрим снова какую-нибудь машину Тьюринга $\mathfrak{T}$ с символами $a_0 = 0$, $a_1 = 1$, $a_2, \ldots, a_m$ и внутренними состояниями $q_0, q_1, \ldots, q_n$. Пусть в начальный момент эта машина имеет конфигурацию, описывающуюся машинным словом $\mathfrak{a}q_i\mathfrak{b}$. Начав работать в этой конфигурации, машина либо через некоторое число тактов работы остановится, придя во внутреннее состояние q_0, либо будет работать вечно. Символом $\mathfrak{T}_{\text{fin}}$ условимся обозначать совокупность машинных слов, описывающих те конфигурации, начиная с которых машина через конечное число тактов работы останавливается, а через $\mathfrak{T}_{\text{inf}}$ обозначим совокупность всех остальных машинных слов.

Определение. *Машина Тьюринга $\mathfrak{T}$ называется универсальной, если множество $\mathfrak{T}_{\text{fin}}$ $\mathfrak{m}$-универсально.*

Прежде всего нам надо убедиться, что универсальные машины существуют. Так как $\mathfrak{m}$-универсальные множества чисел уже были построены в п. 8.1, то нам достаточно доказать лишь истинность следующей теоремы:

Теорема 1. *Если частично рекурсивная функция $f(x)$ имеет своей областью определенности $\mathfrak{m}$-универсальное множество, то вычисляющая эту функцию машина $\mathfrak{T}$ универсальна.*

В самом деле, пусть $\{0, 1, a_2, \ldots, a_m\}$ — внешний алфавит машины $\mathfrak{T}$ и $q_0, q_1, \ldots, q_n$ — ее внутренние состояния. Обозначим через $\boldsymbol{U}$ область определенности функции f и пусть $\boldsymbol{M}_1$ — совокупность всех машинных слов вида $q_1 0 1^x$, а $\boldsymbol{M}_u$ — совокупность тех машинных слов $q_1 0 1^x$, для которых $x \in \boldsymbol{U}$. Ясно, что $\boldsymbol{M}_u$ есть пересечение рекурсивно перечислимой совокупности $\mathfrak{T}_{\text{fin}}$ и рекурсивного множества $\boldsymbol{M}_1$. Но если пересечение рекурсивно перечислимой совокупности и рекурсивного множества $\mathfrak{m}$-универсально, то (см. п. 8.1) совокупность также $\mathfrak{m}$-универсальна, что и требовалось.

Хотя теорема 1 сформулирована для одноместных функций, ясно, что аналогичная теорема верна и для многоместных функций. Известно, что область определенности функции Клини $\boldsymbol{K}(x, y)$ $\mathfrak{m}$-универсальна (п. 8.1). Поэтому машина Тьюринга, вычисляющая $\boldsymbol{K}(x, y)$, универсальна.

Понятие рекурсивного множества нами определялось через понятие рекурсивной функции. Однако его можно определить и более непосредственно в терминах теории машин Тьюринга.

Пусть $\mathfrak{B}$ — какое-нибудь множество слов в произвольном алфавите $\boldsymbol{B}$. Проблемой вхождения для $\mathfrak{B}$ называют проблему отыскания алгоритма, с помощью которого для любого слова $\mathfrak{x}$ в алфавите $\boldsymbol{B}$ можно узнать, входит $\mathfrak{x}$ в $\mathfrak{B}$ или нет. Говорят, что проблема вхождения для множества $\mathfrak{B}$ разрешима на машинах Тьюринга, если существует машина Тьюринга $\mathfrak{T}$, обладающая следующими свойствами:

а) Алфавит $\boldsymbol{B}$ входит во внешний алфавит $\boldsymbol{A}$ машины $\mathfrak{T}$, $\boldsymbol{A}$ содержит символ 0, не входящий в $\boldsymbol{B}$, 1 — символ из $\boldsymbol{B}$.

б) Для каждого слова $\mathfrak{x}$ в алфавите $\boldsymbol{B}$, если $\mathfrak{x} \in \mathfrak{B}$, то машина $\mathfrak{T}$, начав работать в конфигурации $q_1 0\mathfrak{x}$, после конечного числа тактов работы переходит в заключительное состояние $q_0 010 \ldots 0$. Если же $\mathfrak{x} \notin \mathfrak{B}$, то машина $\mathfrak{T}$ из состояния $q_1 0\mathfrak{x}$ переходит в заключительное состояние $q_0 0 \ldots 0$.

Теорема 2. *Проблема вхождения для множества слов $\mathfrak{B}$ тогда и только тогда разрешима на машинах Тьюринга, когда $\mathfrak{B}$ рекурсивно.*

В самом деле, пусть существует машина $\mathfrak{T}$ со свойствами а), б). Обозначим через $F(\mathfrak{x})$ словарную функцию, вычисляемую машиной $\mathfrak{T}$ (в смысле п. 12.2). Обозначим через $\mathfrak{B}_0$ совокупность всевозможных слов в алфавите $\boldsymbol{B}$. Согласно а), б) функция $F(\mathfrak{x})$ всюду определена на $\mathfrak{B}_0$ и $\mathfrak{B}$ есть совокупность решений уравнения $F(\mathfrak{x}) = 1$, лежащих в $\mathfrak{B}_0$. Следовательно, множество $\mathfrak{B}$ рекурсивно.

Обратно, пусть совокупность $\mathfrak{B}$ рекурсивна. Определяем на $\mathfrak{B}_0$ функцию $f(\mathfrak{x})$, полагая ее равной 1 на $\mathfrak{B}$ и Λ вне $\mathfrak{B}$. Так как эта функция рекурсивная, то найдется машина Тьюринга $\mathfrak{T}$, правильно вычисляющая функцию $f(\mathfrak{x})$, а это и значит, что $\mathfrak{T}$ удовлетворяет требованиям а), б).

Пусть $\mathfrak{M}$ — произвольная машина Тьюринга. Говорят, что *проблема остановки машины* $\mathfrak{M}$ разрешима на машинах Тьюринга, если на машинах Тьюринга разрешима проблема вхождения для множества $\mathfrak{M}_{\mathrm{fin}}$, то есть если множество $\mathfrak{M}_{\mathrm{fin}}$ рекурсивно. Так как m-универсальные множества нерекурсивны, то из приведенных определений следует, что проблема остановки универсальной машины Тьюринга неразрешима на машинах Тьюринга.

Дополнения, примеры и задачи

1. Доказать теорему 2 из п. 12.3 о синтезе машин Тьюринга для произвольного внешнего алфавита.

2. Определенные в п. 12.1 машины Тьюринга могут сдвигать ленту на 1 шаг влево или вправо, оставляя содержимое ячеек неизменным, или могут изменять состояние воспринимаемой ячейки, оставляя ленту неподвижной. Можно расширить этот список операций. Машина Тьюринга называется стандартной, если она и при сдвиге ленты может предварительно изменить состояние воспринимаемой ячейки. Программа стандартной машины может быть записана в виде совокупности предписаний вида $q_\alpha a_i \rightarrow q_\beta a_j T_\varepsilon$, где формула $q_\alpha a_i \rightarrow q_\beta a_j T_{-1}$, например, означает, что машина, находясь во внутреннем состоянии q_α и воспринимая символ a_i, заменяет этот символ символом a_j, переходит во внутреннее состояние q_β и сдвигает ленту на 1 шаг влево (если T_1, то сдвигает вправо, если T_0, то оставляет неподвижной). Так как уже на машинах, определенных в п. 12.1, все частично рекурсивные функции вычислимы, то они будут вычислимыми и на стандартных. Показать, что все функции, вычислимые на стандартных машинах, частично рекурсивны.

3. Показать, что каждая частично рекурсивная функция вычислима на машине Тьюринга, способной выполнять лишь предписания вида

$$q_\alpha a_i \rightarrow q_\alpha a_i T_\varepsilon, \quad q_\alpha a_i \rightarrow q_\beta a_i T_0, \quad q_\alpha a_i \rightarrow q_\alpha a_j T_0 \ (\varepsilon = \pm 1).$$

4. Стандартная машина Тьюринга с внешним алфавитом 0,1 называется нестирающей, если она способна выполнять лишь предписания вида

$$q_\alpha 0 \rightarrow q_\beta a T_\varepsilon, \quad q_\alpha 1 \rightarrow q_\beta 1 T_\varepsilon \quad (a = 0,\ 1;\ \varepsilon = 0,\ \pm 1),$$

т. е. если она может вписать 1 в пустую ячейку, но не может удалить символ 1, если он уже вписан в ячейку. Показать, что при подходящем кодировании чисел любая частично рекурсивная функция вычислима на подходящей нестирающей машине. В частности, существуют универсальные нестирающие машины (Ван Хао [11], обобщение и другое доказательство см. Зыкин [37]).

5. Пусть задана машина Тьюринга $\mathfrak{T}$ с внешним алфавитом $a_0, a_1, \ldots, a_m$ и внутренними состояниями $q_0, q_1, \ldots, q_n$. Системой кодов условимся называть пару словарных функций $\mathrm{cod}(\mathfrak{x}, \mathfrak{y})$, $\mathrm{dec}(\mathfrak{x}, \mathfrak{y})$, удовлетворяющих следующим условиям: обе функции общерекурсивные, определенные в объединенном алфавите $\{a_0, \ldots, a_m; q_0, \ldots, q_n\}$; б) если $\mathfrak{y}$ — слово в алфавите $\{a_1, \ldots, a_m\}$, то $\mathrm{cod}(\mathfrak{x}, \mathfrak{y})$ — некоторое машинное слово, если же $\mathfrak{y}$ — заключительное машинное слово, то $\mathrm{dec}(\mathfrak{x}, \mathfrak{y})$ — слово в алфавите $\{a_1, \ldots, a_m\}$. Говорят, что машина $\mathfrak{T}$ вычисляет частичную словарную функцию $f(\mathfrak{x})$, заданную в алфавите $\{a_1, \ldots, a_m\}$ при коде с индексом $\mathfrak{n}$, если машина, начав работать в конфигурации $\mathrm{cod}(\mathfrak{n}, \mathfrak{x})$, или остановится в конфигурации $\mathfrak{z}$, для которой $\mathrm{dec}(\mathfrak{n}, \mathfrak{z}) = f(\mathfrak{x})$, либо будет работать вечно, если $f(\mathfrak{x})$ не определено.

Машина $\mathfrak{T}$ называется *строго универсальной*, если для нее существует такая система cod $(\mathfrak{x}, \mathfrak{y})$, dec $(\mathfrak{x}, \mathfrak{y})$, что каждая частично рекурсивная числовая функция $f(x)$ вычислима на $\mathfrak{T}$ при коде с подходящим индексом.

Показать, что каждая строго универсальная машина универсальна. Машина, вычисляющая в обычном смысле функцию Клини $K(x, y)$, является строго универсальной.

6. Каждая машина Тьюринга $\mathfrak{T}$, имеющая лишь одно внутреннее состояние, отличное от заключительного, имеет рекурсивное множество $\mathfrak{T}_{\mathfrak{fin}}$ и потому не может быть универсальной (Шеннон [117]).

7. Используя простое множество, построенное в п. 8.3, показать, что из нерекурсивности множества $\mathfrak{T}_{\mathfrak{fin}}$ еще не следует универсальность машины $\mathfrak{T}$.

8. Показать, что каждая частично рекурсивная функция при подходящем кодировании натуральных чисел вычислима на машине Тьюринга, имеющей внутренние состояния q_0, q_1, q_2 (то есть имеющей лишь два внутренних состояния, отличных от заключительного) и достаточно большой внешний алфавит (Шеннон [117]).

9. Обычно принимается (см. Шеннон [117]), что «сложность» машины Тьюринга равна произведению числа символов ее внешнего алфавита на число внутренних состояний, отличных от заключительного. Значительный интерес вызвала задача построения универсальных машин минимальной сложности. К началу 1963 г. последними результатами в этом направлении были теорема Ватанабе [13] о существовании универсальной машины с пятью символами и шестью состояниями и теорема Триттера [100] о существовании универсальной машины с четырьмя символами и шестью состояниями (отличными от заключительного).

10. Пусть $f(x_1, \ldots, x_s)$ — общерекурсивная функция и $\mathfrak{T}$ — машина Тьюринга с внешним алфавитом $\{0, 1, \alpha\}$ и внутренними состояниями $q_0, q_1, \ldots, q_n$, α — символ пустой ячейки, q_0 — заключительное состояние. Условимся говорить, что машина $\mathfrak{T}$ вычисляет функцию f, если для любых натуральных $x_1, \ldots, x_s$

$$q_1\alpha \{x_1\}_2 \alpha \ldots \alpha \{x_s\}_2 \vdash \ldots q_0\alpha \{f(x_1, \ldots, x_s)\}_2 \alpha \ldots \alpha,$$

где $\{x\}_2$ — двоичная запись числа x. Мы предполагаем, что машина может в процессе вычислений надстраивать пустые ячейки, но предполагаем также, что машина не может отбрасывать никаких ячеек. Поэтому длина машинных слов в процессе вычислений не убывает и заключительные слова — это слова максимальной длины. Пусть $l_f(x_1, \ldots, x_s)$ — длина заключительного слова, получившегося при вычислении значения $f(x_1, \ldots, x_s)$ на машине $\mathfrak{T}$, уменьшенная на 1. Это длина ленты, необходимой машине $\mathfrak{T}$ для вычисления $f(x_1, \ldots, x_s)$.

Как измерять «сложность» вычислимой функции? Один из возможных ответов на этот вопрос указан Ричи [85]. Рекурсивные функции разбиваются на классы $F_0 \subset F_1 \subset \ldots$ по мере возрастания их сложности следующим образом. В класс F_0 относим все линейные функции. Далее для каждого l в класс F_{l+1} относим те функции

$f(x_1, \ldots, x_s)$, для которых существует вычисляющая их машина Тьюринга $\mathfrak{T}$ такая, что $l_f \in F_l$. В статье Ричи показано, что объединение всех классов F_l в точности совпадает с классом функций, элементарных в смысле Кальмара — Чиллага [38] (см. задачи 4, 5 из § 6). Пусть $\psi_1(x) = 2^x$, $\psi_{l+1}(x) = \psi_1(\psi_l(x))$. Тогда $\psi_l \in F_l$, $\psi_{l+1} \notin F_{l+1}$ и потому классы F_l различны.

Аналогичная идея для разбиения функций в классы по степеням сложности была развита также Кливом [40].

§ 13. ПРИЛОЖЕНИЯ

Изложенная выше теория машин Тьюринга позволяет просто решить ряд алгоритмических проблем алгебры, логики, арифметики и других математических дисциплин. Некоторые из этих проблем в качестве иллюстрации будут рассмотрены в данном параграфе. Приложения к теории чисел будут изложены в § 16.

13.1. Проблема равенства слов в полугруппах. В качестве одного из приложений теории машин Тьюринга мы хотим сейчас рассмотреть проблемы равенства (или эквивалентности) слов в полугруппах, заданных определяющими соотношениями.

Совокупность элементов, рассматриваемая вместе с некоторой определенной на ней бинарной ассоциативной операцией, называется ассоциативной системой или *полугруппой*. Например, полугруппой является совокупность всех натуральных чисел вместе с операцией сложения или вместе с операцией умножения.

Основную операцию в произвольной полугруппе обычно называют умножением и обозначают точкой. Вместо $a \cdot b$ обычно пишут ab. Так как распределение скобок в произведениях нескольких сомножителей: $((ab)c)d$, $a(b(cd))$ на величину этих произведений в полугруппах не влияет, то при записи таких произведений скобки опускают.

Согласно п. 1.3 система элементов $c_1, \ldots, c_p$ некоторой полугруппы $\mathfrak{C}$ называется системой порождающих для $\mathfrak{C}$, если каждый элемент c полугруппы $\mathfrak{C}$ может быть представлен в форме

$$c = c_{i_1} c_{i_2} \ldots c_{i_s} \qquad (i_k = 1, 2, \ldots, p),$$

где множители могут и повторяться. Например, числа вида 2^m образуют полугруппу относительно умножения с одним порождающим элементом 2.

В качестве другого примера рассмотрим совокупность всех непустых слов какого-нибудь алфавита $\boldsymbol{C} = \{c_1, \ldots \ldots, c_p\}$. Эта совокупность является полугруппой относительно обычной операции перемножения слов: $\mathfrak{a} \cdot \mathfrak{b} = \mathfrak{a}\mathfrak{b}$. Однобуквенные слова $c_1, \ldots, c_p$, очевидно, будут порождающими для этой полугруппы, называющейся *свободной полугруппой* с свободными порождающими $c_1, \ldots, c_p$.

В алгебре важное место занимает способ задания полугрупп посредством определяющих соотношений. Сущность его состоит в следующем.

Задаем некоторый алфавит $\boldsymbol{C} = \{c_1, \ldots, c_p\}$ и конечную совокупность формальных равенств вида

$$\mathfrak{c}_1 \equiv \mathfrak{d}_1, \quad \mathfrak{c}_2 \equiv \mathfrak{d}_2, \ \ldots, \ \mathfrak{c}_l \equiv \mathfrak{d}_l, \tag{1}$$

где $\equiv$ — особый знак, а $\mathfrak{c}_1, \mathfrak{d}_1, \ldots, \mathfrak{c}_l, \mathfrak{d}_l$ — произвольные слова в алфавите $\{c_1, \ldots, c_p\}$. В течение некоторого времени словами далее будем называть лишь слова в алфавите $\{c_1, \ldots, c_p\}$ и делать специальные оговорки в противном случае.

Левым элементарным преобразованием некоторого слова $\mathfrak{c}$, отвечающим соотношению $\mathfrak{c}_i \equiv \mathfrak{d}_i$, будем называть подстановку в слово $\mathfrak{c}$ слова $\mathfrak{d}_i$ вместо какого-нибудь вхождения слова $\mathfrak{c}_i$. Аналогично, *правым элементарным преобразованием*, отвечающим соотношению $\mathfrak{c}_i \equiv \mathfrak{d}_i$, называется подстановка в слово $\mathfrak{c}$ слова $\mathfrak{c}_i$ вместо какого-нибудь вхождения слова $\mathfrak{d}_i$ в $\mathfrak{c}$.

Если слово $\mathfrak{c}$ содержит несколько вхождений слова $\mathfrak{c}_i$, то над $\mathfrak{c}$ можно выполнить по произволу одно из нескольких левых элементарных преобразований, отвечающих соотношению $\mathfrak{c}_i \equiv \mathfrak{d}_i$. Если $\mathfrak{c}_i$ в $\mathfrak{c}$ не входит, то над $\mathfrak{c}$ соответствующие левые элементарные преобразования не выполнимы.

Левые и правые элементарные преобразования, отвечающие определяющим равенствам (1), называются просто элементарными преобразованиями. Условимся писать $\mathfrak{c} \to \mathfrak{d}$, если слово $\mathfrak{c}$ получается каким-либо элементарным преобразованием из $\mathfrak{c}$, и условимся писать $\mathfrak{c} \equiv \mathfrak{d}$, если существуют такие слова $\mathfrak{x}_1, \ldots, \mathfrak{x}_l$, что

$$\mathfrak{c} \to \mathfrak{x}_1 \to \ldots \to \mathfrak{x}_l \to \mathfrak{d}, \tag{2}$$

или если $\mathfrak{c} = \mathfrak{d}$.

Новое определение отношения $\equiv$ не противоречит определяющим соотношениям (1), так как в силу (1) $\mathfrak{c}_i \to \mathfrak{d}_i$ и потому $\mathfrak{c}_i \equiv \mathfrak{d}_i$.

Из приведенных определений непосредственно видно, что отношение $\equiv$ рефлексивно, транзитивно, симметрично и согласовано с операцией умножения, то есть

$$\mathfrak{x} \equiv \mathfrak{x}_1 \& \mathfrak{y} \equiv \mathfrak{y}_1 \Rightarrow \mathfrak{x}\mathfrak{y} \equiv \mathfrak{x}_1\mathfrak{y}_1. \qquad (3)$$

Отношение $\equiv$ будем для краткости называть эквивалентностью. Совокупность всех слов, эквивалентных данному слову $\mathfrak{c}$, обозначим через $[\mathfrak{c}]$ и назовем смежным классом по модулю $\equiv$. Ввиду изложенного соотношение $\mathfrak{x} \equiv \mathfrak{y}$ равносильно «точному» равенству $[\mathfrak{x}] = [\mathfrak{y}]$.

Вводим теперь операцию умножения смежных классов, полагая по определению

$$[\mathfrak{x}] \cdot [\mathfrak{y}] = [\mathfrak{x}\mathfrak{y}]. \qquad (4)$$

В силу (3) так определенная операция умножения является однозначной на совокупности всех смежных классов. Поскольку для произвольных слов $\mathfrak{a}$, $\mathfrak{b}$, $\mathfrak{c}$ из (4) следует

$$([\mathfrak{a}][\mathfrak{b}])[\mathfrak{c}] = [\mathfrak{a}\mathfrak{b} \cdot \mathfrak{c}] = [\mathfrak{a} \cdot \mathfrak{b}\mathfrak{c}] = [\mathfrak{a}]([\mathfrak{b}][\mathfrak{c}]),$$

то операция умножения смежных классов ассоциативна и, следовательно, совокупность всех смежных классов по модулю $\equiv$ является полугруппой, которую мы будем обозначать через $\mathfrak{C}_{(1)}$.

Для любого слова $\mathfrak{c} = c_{i_1}c_{i_2} \ldots c_{i_s}$ из (4) следует

$$[\mathfrak{c}] = [c_{i_1} \ldots c_{i_s}] = [c_{i_1}] \ldots [c_{i_s}].$$

Поэтому элементы $[c_1], \ldots, [c_p]$ образуют порождающую систему для полугруппы $\mathfrak{C}_{(1)}$ и полугруппа $\mathfrak{C}_{(1)}$ обычно называется *полугруппой, заданной порождающими* $c_1, \ldots, c_p$ *и определяющими соотношениями* (1). Полугруппы, задаваемые этим путем, называются конечно определенными полугруппами или ассоциативными исчислениями.

Говорят, что слово $\mathfrak{c}$ *представляет* элемент $[\mathfrak{c}]$ полугруппы $\mathfrak{C}_{(1)}$. Так как основной способ задать элемент полугруппы — это задать его представителя, то естественно возникает следующая основная

Проблема равенства для ассоциативных исчислений. *Для заданного ассоциативного исчисления указать алгоритм, посредством которого для любых двух слов* $\mathfrak{a}$, $\mathfrak{b}$ *можно было бы сказать, представляют ли они один и тот же элемент полугруппы, то есть эквивалентны ли они в заданном исчислении.*

В точных терминах эта проблема может быть сформулирована так: для заданного ассоциативного исчисления является ли рекурсивной совокупность всех пар слов, эквивалентных в данном исчислении?

Небольшое изучение вопроса приводит к выводу, что, во всяком случае, справедлива

Теорема 1. *В каждом ассоциативном исчислении совокупность пар эквивалентных слов рекурсивно перечислима.*

В самом деле, для каждого данного слова $\mathfrak{a}$ легко найти все слова, получаемые из него одним элементарным преобразованием, двумя последовательными элементарными преобразованиями и т. д. Производя сначала по одному элементарному преобразованию над однобуквенными словами, затем не более чем по два последовательных элементарных преобразования над словами длины, не превосходящей 2, и т. д., мы постепенно перечислим все пары эквивалентных слов. Для оформления этого рассуждения в точное доказательство достаточно на множестве пар слов определить надлежащие операции, как это неоднократно делалось, и сослаться на теорему о порожденных совокупностях из п. 4.3.

Из теоремы 1 непосредственно следует, что в каждом ассоциативном исчислении каждый смежный класс рекурсивно перечислим. Спрашивается, а не будут ли все смежные классы во всех ассоциативных исчислениях просто рекурсивными? Ясно, что если в каком-нибудь ассоциативном исчислении проблема тождества решается положительно, то все смежные классы в этом исчислении рекурсивны.

Пользуясь близостью элементарных преобразований слов в ассоциативных исчислениях к преобразованиям машинных слов в машинах Поста — Тьюринга, Пост [73] и Марков [55] почти одновременно заметили, как можно построить ассоциативное исчисление, обладающее нерекурсивными смежными классами и потому имеющее

неразрешимую проблему тождества. Мы сейчас изложим один из простейших способов построения ассоциативных исчислений, элементарные преобразования которых непосредственно имитируют преобразования машинных слов в машинах Тьюринга.

Пусть задана машина Тьюринга с символами $a_0, a_1, \ldots$ $\ldots, a_m$, состояниями $q_0, q_1, \ldots, q_n$ и совокупностью $\mathfrak{T}$ команд. Строим ассоциативное исчисление $A(\mathfrak{T})$ следующим образом. Алфавит $A(\mathfrak{T})$ состоит из символов $a_0, a_1, \ldots, a_m$, $q_0, q_1, \ldots, q_n$ и дополнительного символа v. Определяющие соотношения исчисления $A(\mathfrak{T})$ выписываются при помощи команд $\mathfrak{T}$. Ниже слева указаны команды из $\mathfrak{T}$, а справа выписаны соответствующие формальные равенства, которые вносятся в список определяющих соотношений для $A(\mathfrak{T})$:

$$q_\alpha a_i \to q_\beta a_j \qquad q_\alpha a_i \equiv q_\beta a_j, \tag{5}$$

$$q_\alpha a_i \to q_\beta L \qquad a_k q_\alpha a_i \equiv q_\beta a_k a_i \quad (k = 0, 1, \ldots, m), \tag{6}$$

$$q_\alpha a_i \to q_\beta R \qquad q_\alpha a_i \equiv a_i q_\beta. \tag{7}$$

Добавляя к этим соотношениям еще соотношение

$$a_0 v \equiv v, \tag{8}$$

получим полный список определяющих соотношений исчисления $A(\mathfrak{T})$.

Теорема 2. *Пусть машина Тьюринга $\mathfrak{T}$ имеет символы $a_0, a_1, \ldots, a_m$ и состояния $q_0, q_1, \ldots, q_n$, $A(\mathfrak{T})$ — построенное выше ассоциативное исчисление с определяющими соотношениями* (5) — (8), $\mathfrak{a}$, $\mathfrak{b}$, $\mathfrak{c}$, $\mathfrak{d}$ — *произвольные слова в алфавите $a_0, a_1, \ldots, a_m$, из которых слово $\mathfrak{d}$ либо пусто, либо оканчивается буквой, отличной от a_0. Для того чтобы в исчислении $A(\mathfrak{T})$ было истинно соотношение*

$$\mathfrak{a} q_w a_l \mathfrak{b} v \equiv \mathfrak{c} q_0 a_p \mathfrak{d} v,$$

необходимо и достаточно, чтобы машина $\mathfrak{T}$ из конфигурации $\mathfrak{a} q_w a_l \mathfrak{b}$ через конечное число тактов работы и без надстраивания ячеек слева переходила в конфигурацию $\mathfrak{c} q_0 a_p \mathfrak{d} a_0^x$.

Достаточность условий совершенно очевидна. Действительно, пусть

$$\mathfrak{a} q_w a_l \mathfrak{b} \to \mathfrak{m}_1 \to \ldots \to \mathfrak{m}_t = \mathfrak{c} q_0 a_p \mathfrak{d} a_0^x$$

— последовательные конфигурации, принимаемые работающей машиной $\mathfrak{T}$. По условию, конфигурация $\mathfrak{m}_{k+1}$ возникает из конфигурации $\mathfrak{m}_k$ путем выполнения некоторой команды из совокупности $\mathfrak{T}$. Эта команда имеет один из видов (5), (6), (7). Рассмотрим самый сложный случай: пусть слово $\mathfrak{m}_k$ переходит в слово $\mathfrak{m}_{k+1}$ в результате выполнения команды вида (7), сопровождаемой надстраиванием ячейки справа. Это значит, что $\mathfrak{m}_k = \mathfrak{a}' q_\alpha a_i$, $\mathfrak{m}_{k+1} = \mathfrak{a}' a_i q_\beta a_0$. В слове $\mathfrak{m}_k v$ заменяем согласно (8) букву v словом $a_0 v$, затем, заменяя в слове $\mathfrak{m}_k a_0 v$ согласно (7) подслово $q_\alpha a_i$ через $a_i q_\beta$, получим слово $\mathfrak{m}_{k+1} v$, то есть из $\mathfrak{m}_k \to \mathfrak{m}_{k+1}$ следует $\mathfrak{m}_k v \equiv \mathfrak{m}_{k+1} v$, что и требовалось.

Докажем теперь необходимость условий теоремы 2. Пусть

$$\mathfrak{a} q_w a_l \mathfrak{b} v = \mathfrak{x}_0 \to \mathfrak{x}_1 \to \ldots \to \mathfrak{x}_t = \mathfrak{c} q_0 a_p \mathfrak{d} v \tag{9}$$

— слова, получаемые последовательно элементарными преобразованиями, отвечающими соотношениям (5) — (8). Не меняя начального и конечного слов цепочки (9), мы хотим ее преобразовать в такую цепочку, для которой доказываемое утверждение будет очевидным.

Из формы элементарных преобразований (5) — (8) следует, что в преобразуемом слове они не меняют ни числа вхождений буквы v, ни числа вхождений каждой буквы вида q_α. Поэтому каждое слово в цепочке (9) имеет вид

$$\mathfrak{x}_k = \mathfrak{a}_k q_{w_k} a_{l_k} \mathfrak{b}_k v,$$

где $\mathfrak{a}_k$, $\mathfrak{b}_k$ — некоторые слова в алфавите $a_0, a_1, \ldots, a_m$.

Пусть в двух последовательных преобразованиях $\mathfrak{x}_k \to \mathfrak{x}_{k+1} \to \mathfrak{x}_{k+2}$ из (9) второе преобразование есть правое v-преобразование $v \to a_0 v$. Тогда в более подробной записи этот участок цепочки (9) имеет вид

$$\mathfrak{a}' q_\alpha \mathfrak{b}' v \to \mathfrak{a}'' q_\beta \mathfrak{b}'' v \to \mathfrak{a}'' q_\beta \mathfrak{b}'' a_0 v$$

и потому вместо цепочки (9) можно рассматривать цепочку

$$\ldots \to \mathfrak{a}' q_\alpha \mathfrak{b}' v \to \mathfrak{a}' q_\alpha \mathfrak{b}' a_0 v \to \mathfrak{a}'' q_\beta \mathfrak{b}'' a_0 v \to \ldots \tag{10}$$

Переход от цепочки (9) к цепочке (10) условимся называть сдвигом правого v-преобразования $\mathfrak{x}_{k+1} \to \mathfrak{x}_{k+2}$ влево.

Аналогично, если в отрезке $\mathfrak{x}_k \to \mathfrak{x}_{k+1} \to \mathfrak{x}_{k+2}$ преобразование $\mathfrak{x}_k \to \mathfrak{x}_{k+1}$ есть левое v-преобразование $a_0 v \to$

$\to v$, то этот отрезок имеет вид

$$\mathfrak{a}'q_\alpha\mathfrak{b}'a_0v \to \mathfrak{a}'q_\alpha\mathfrak{b}'v \to \mathfrak{a}''q_\beta\mathfrak{b}''v$$

и вместо цепочки (9) можно рассматривать цепочку

$$\ldots \to \mathfrak{a}'q_\alpha\mathfrak{b}'a_0v \to \mathfrak{a}''q_\beta\mathfrak{b}''a_0v \to \mathfrak{a}''q_\beta\mathfrak{b}''v \to \ldots \quad (11)$$

Переход от (9) к (11) будем называть сдвигом левого v-преобразования $\mathfrak{x}_k \to \mathfrak{x}_{k+1}$ вправо.

Сдвигая в цепочке (9) все правые v-преобразования влево, а все левые v-преобразования вправо достаточное число раз, получим цепочку вида

$$\mathfrak{a}q_wa_l\mathfrak{b}v \to \ldots \to \mathfrak{a}q_wa_l\mathfrak{b}a_0^yv \to \ldots$$
$$\ldots \to \mathfrak{c}q_0a_p\mathfrak{d}a_0^zv \to \ldots \to \mathfrak{c}q_0a_p\mathfrak{d}v,$$

где участок

$$\mathfrak{a}q_wa_l\mathfrak{b}a_0^yv \to \mathfrak{y}_1 \to \ldots \to \mathfrak{y}_s = \mathfrak{c}q_0a_p\mathfrak{d}a_0^zv \quad (12)$$

преобразований вида (8) уже не содержит.

Последнее слово цепочки (12) содержит заключительный символ q_0, не входящий в левые слова соотношений (5), (6), (7). Поэтому преобразование $\mathfrak{y}_{s-1} \to \mathfrak{y}_s$ не может быть правым. Допустим, что цепочка (12) содержит правые преобразования. Пусть $\mathfrak{y}_k \to \mathfrak{y}_{k+1}$ — последнее правое преобразование в ней, отвечающее соотношению (u) ($u = 5$, 6, 7). Слово $\mathfrak{y}_{k+1}$ пусть содержит подслово $q_\alpha a_i$. Преобразование $\mathfrak{y}_{k+1} \to \mathfrak{y}_{k+2}$ по предположению левое. Но среди соотношений (5), (6), (7) есть лишь одно соотношение, левая часть которого есть $q_\alpha a_i$. Следовательно, это соотношение должно совпадать с упомянутым соотношением (u). Таким образом, преобразования $\mathfrak{y}_k \to \mathfrak{y}_{k+1}$ и $\mathfrak{y}_{k+1} \to \mathfrak{y}_{k+2}$ взаимно обратны, $\mathfrak{y}_k = \mathfrak{y}_{k+2}$ и цепочку (12) можно сократить до цепочки

$$\mathfrak{y}_0 \to \ldots \to \mathfrak{y}_k \to \mathfrak{y}_{k+3} \to \ldots \to \mathfrak{y}_s. \quad (13)$$

Продолжая этот процесс дальше, мы получим цепочку вида (12), в которой все преобразования левые, отвечающие соотношениям (5), (6), (7). Но ведь эти соотношения были так подобраны, чтобы, выполняя левое преобразование, отвечающее одному из указанных соотношений, над какой-нибудь конфигурацией $\mathfrak{m}$ машины $\mathfrak{T}$, мы получали конфигурацию $\mathfrak{m}^{(1)}$ следующего состояния машины $\mathfrak{T}$.

Следовательно, машина $\mathfrak{T}$, начав работать в конфигурации $\mathfrak{a}q_w a_l \mathfrak{b} a_0^y$, закончит работу в конфигурации $\mathfrak{c} q_0 a_p \mathfrak{d} a_0^z$. Последняя буква слова $\mathfrak{b}$ отлична от a_0. Поэтому машина $\mathfrak{T}$, начав работать в конфигурации $\mathfrak{a}q_w a_l \mathfrak{b}$, закончит работу в конфигурации вида $\mathfrak{c} q_0 a_p \mathfrak{d} a_0^x$, что и требовалось.

Теорема 3 (Поста — Маркова). *Существует ассоциативное исчисление с алгоритмически неразрешимой проблемой равенства.*

В п. 6.3 построена частично рекурсивная функция $E(x)$, принимающая лишь значения 0, 1 и не имеющая рекурсивных доопределений. Пусть $\boldsymbol{M}_i$ $(i = 0, 1)$ — совокупность решений уравнения $E(x) = i$. В п. 6.3 показано, что множества $\boldsymbol{M}_i$ рекурсивно перечислимые, но не рекурсивные. Обозначим через $\mathfrak{T}$ машину Тьюринга с символами 0, 1 и подходящими состояниями $q_0, q_1, \ldots, q_n$, правильно вычисляющую функцию $E(x)$, то есть переходящую из конфигурации $q_1 01^x$ в конфигурацию вида $q_0 01^{E(x)} 0^s$ после конечного числа тактов работы без надстраивания ячеек слева. Согласно теореме 2 это означает, что в ассоциативной системе $A(\mathfrak{T})$ для любого натурального x

$$q_1 01^x v \equiv q_0 01^i v \iff x \in \boldsymbol{M}_i$$

или

$$x \in \boldsymbol{M}_i \iff q_1 01^x v \in [q_0 01^i v] \quad (i = 0, 1),$$

где $[\mathfrak{x}]$ обозначает совокупность слов исчисления $A(\mathfrak{T})$, эквивалентных слову $\mathfrak{x}$. Алфавитный номер (п. 11.1) слова $q_1 01^x v$ будет некоторой общерекурсивной функцией $\varphi(x)$. Обозначая через $\boldsymbol{P}_i$ $(i = 0, 1)$ совокупность алфавитных номеров слов из класса $[q_0 01^i v]$, получим

$$x \in \boldsymbol{M}_i \iff \varphi(x) \in \boldsymbol{P}_i \quad (i = 0, 1). \tag{14}$$

Мы видим, следовательно, что нерекурсивное множество $\boldsymbol{M}_0$ $\boldsymbol{m}$-сводится к множеству $\boldsymbol{P}_0$. Поэтому множество $\boldsymbol{P}_0$ не рекурсивно и в исчислении $A(\mathfrak{T})$ проблема эквивалентности произвольного слова слову $q_0 v$ алгоритмически неразрешима.

На самом деле соотношения (14) дают несколько больше, чем утверждает теорема 3. Именно, соотношения (14) означают, что пара множеств $\boldsymbol{M}_0$, $\boldsymbol{M}_1$ $\boldsymbol{m}$-сводится к паре $\boldsymbol{P}_0$, $\boldsymbol{P}_1$. Так как пара $\boldsymbol{M}_0$, $\boldsymbol{M}_1$ $\boldsymbol{m}$-универсальна, а множества

$\boldsymbol{P}_0$, $\boldsymbol{P}_1$ рекурсивно перечислимы, то пара $\boldsymbol{P}_0$, $\boldsymbol{P}_1$ также $\boldsymbol{m}$-универсальна. Следовательно, ассоциативное исчисление $A(\mathfrak{T})$ обладает парой эффективно неотделимых смежных классов $[q_0 v]$ и $[q_0 1 v]$.

Выше мы строили ассоциативное исчисление с неразрешимой проблемой равенства, совершенно не заботясь о том, насколько сложным это исчисление окажется. В первом приближении можно принять, что сложность исчисления характеризуется числом порождающих и числом определяющих соотношений. Простейшие замечания (см. задачу 12, § 13) показывают, что если существует ассоциативное исчисление с неразрешимой проблемой равенства, имеющее p порождающих и l определяющих соотношений, то существует и ассоциативное исчисление с неразрешимой проблемой равенства, имеющее то же число l определяющих соотношений и лишь 2 порождающих. Возникает задача нахождения ассоциативного исчисления с неразрешимой проблемой равенства и наименьшим числом определяющих соотношений. Есть основания полагать (см. С. И. Адян [3]), что ассоциативные исчисления с одним определяющим соотношением имеют разрешимую проблему равенства. С другой стороны, Г. С. Цейтин [112] показал, что ассоциативное исчисление с порождающими a, b, c, d, e и семью определяющими соотношениями

$$ac \equiv ca, \qquad ad \equiv da, \qquad bc \equiv cb, \qquad bd \equiv db,$$
$$eca \equiv ce, \qquad edb \equiv de, \qquad cca \equiv ccae$$

имеет неразрешимую проблему равенства слов. Ассоциативное исчисление, имеющее неразрешимую проблему равенства и менее семи определяющих соотношений, пока неизвестно.

Ассоциативное исчисление называется *групповым* (иногда — инверсивным) исчислением, если его порождающие можно разбить на пары $a_1, b_1, \ldots, a_p, b_p$ так, чтобы среди определяющих соотношений заведомо встречались соотношения

$$a_1 b_1 \equiv \Lambda, \ \ldots, \ a_p b_p \equiv \Lambda,$$

где Λ — пустое слово. Полугруппа, определяемая групповым исчислением, является группой, в которой элементы $[a_i]$ и $[b_i]$ взаимно обратны.

Задание групп при помощи групповых исчислений, то есть при помощи порождающих и определяющих соотношений, играет большую роль как в самой теории групп, так и в некоторых ее важных приложениях. Поэтому уже в начале XX столетия в теории групп стала известной проблема равенства слов. Однако она оказалась весьма трудной и решение ее было получено только в 1952 г. П. С. Новиковым [66]. Другие ее решения позже были найдены В. Буном [9], Д. Бриттоном [8], Г. Хигманом [110]. Все эти решения достаточно сложны и потому здесь не излагаются.

Конечно, в связи с ассоциативными и групповыми исчислениями возникло много различных проблем алгоритмического характера. Общий обзор их дан в работах А. А. Маркова [55], С. И. Адяна [2], а также М. Рабина [79]. Как правило, решение этих проблем осуществляется тем или иным сведением их к проблеме равенства слов в некотором вспомогательном ассоциативном или групповом исчислении с заведомо неразрешимой проблемой равенства слов.

13.2. Тождественно истинные формулы исчисления предикатов 1-й ступени. Основным формальным логическим языком является исчисление предикатов 1-й ступени. Наша цель — доказать теорему Чёрча о том, что совокупность всех тождественно истинных формул исчисления предикатов 1-й ступени нерекурсивна. Напомним некоторые определения.

Рассматривается особое двухэлементное множество ε, элементы которого называются «истиной» и «ложью» и обозначаются соответственно *И* и *Л*. На множестве ε определяются операции конъюнкции $\&$, дизъюнкции $\vee$, импликации $\rightarrow$ и отрицания $\neg$ обычным образом: $a \& b$ имеет значение *И* для $a = И$, $b = И$ и $a \& b = Л$ для остальных значений a, b; далее,

$$a \vee b = Л \iff a = b = Л,$$

$$a \rightarrow b = Л \iff a = И,\ b = Л,$$

$$\neg a = И \iff a = Л.$$

Если M — произвольное непустое множество, то n-местная функция $P(x_1, \ldots, x_n)$, определенная на M со

значениями в $\{И, Л\}$, называется n-местным *предикатом* на $\boldsymbol{M}$. Для двуместных предикатов P вместо записи $P(x, y)$ обычно употребляют более короткую запись xPy. Например, для обычного отношения $<$, определенного на множестве натуральных чисел, имеем

$$1<2=И, \quad 3<2=Л, \quad 1<2 \vee 2<3=И.$$

Символ равенства $=$ будет рассматриваться также как символ двуместного предиката, определенного на любом множестве.

Алфавит $\boldsymbol{A}$ рассматриваемого исчисления предикатов состоит из символов $\&$, $\vee$, $\rightarrow$, $\neg$, $\exists$, $\forall$, (,), $\cdot$, $=$, P, F, α, β, γ. Слова вида $(F\alpha \ldots \alpha\beta \ldots \beta) = (F\alpha^m\beta^n)$ и $(P\alpha^m\beta^n)$ будем обозначать через F_n^m, P_n^m и называть соответственно m-местными функциональным и предикатным символами. Слова $x_i = (\gamma^{i+1})$ $(i = 0, 1, \ldots)$ будут называться предметными символами.

Задать «значение» какого-нибудь предметного символа x_i в некотором множестве $\boldsymbol{M}$ — значит поставить этому символу в соответствие какой-либо элемент из $\boldsymbol{M}$. Задать значение какого-нибудь функционального символа F_n^m на множестве $\boldsymbol{M}$ — значит поставить ему в соответствие некоторую m-местную операцию, определенную на $\boldsymbol{M}$. Аналогично, задать «значение» предикатного символа P_n^m — значит поставить ему в соответствие какой-либо m-местный предикат на $\boldsymbol{M}$.

Слова в алфавите $\boldsymbol{A}$, имеющие определенный ниже специальный вид, называются термами и формулами. Понятие терма уже было введено в § 1: термами называются предметные символы x_i, а также выражения вида $F_n^m(\mathfrak{a}_1, \ldots, \mathfrak{a}_m)$, где $\mathfrak{a}_1, \ldots, \mathfrak{a}_m$ — термы меньшей длины. Например, термами являются слова

$$F_1^2(F_1^1(x_0), x_1), \quad F_0^3(x_1, x_2, x_1),$$

тогда как слово $P_1^1(x_0)$ термом не является.

Пусть $\mathfrak{a}$ — некоторый терм, и пусть значения всех входящих в запись $\mathfrak{a}$ функциональных и предметных символов заданы на некотором множестве $\boldsymbol{M}$. Тогда, выполняя над значениями предметных переменных последовательно операции, указанные в записи терма, получим в результате

некоторый элемент из M, который и называется значением терма $\mathfrak{a}$ при заданных значениях предметных и функциональных переменных. Например, рассмотрим терм

$$F_1^2(x_0,\ F_0^2(x_0,\ x_1)).$$

Пусть значениями символов x_0, x_1, F_1^2, F_0^2 будут соответственно числа 3, 2 и операции $+$, $\times$, определенные на множестве натуральных чисел. Тогда значением терма будет

$$3+(3\times 2)=9.$$

Если в каком-нибудь терме $\mathfrak{a}$ значения всех функциональных и некоторых предметных символов фиксированы, то значение этого терма будет функцией от значений остальных предметных символов, участвующих в записи терма. Функции такого рода называются термальными или представляющимися термами.

Перейдем к определению понятия формулы.

а) Если $\mathfrak{a}_1, \ldots, \mathfrak{a}_m$ — термы, то слова вида

$$P_n^m(\mathfrak{a}_1, \ldots, \mathfrak{a}_m),\ \mathfrak{a}_{i_1}=\mathfrak{a}_{i_2}$$

называются первичными формулами. Пусть на некотором множестве M заданы значения предикатного символа P_n^m и значения всех предметных и функциональных символов, встречающихся в записи термов $\mathfrak{a}_1, \ldots, \mathfrak{a}_m$. Тогда значения термов $\mathfrak{a}_i$ будут вполне определенными элементами $a_1, \ldots, a_m$ из M и выражение $\widetilde{P}_n^m(a_1, \ldots, a_m)$ ($\widetilde{P}$ — значение символа P) будет равно $И$ или $Л$. Это и будет значением рассматриваемой первичной формулы.

Предметный символ x_i называется связанным в слове $\mathfrak{b}$, если слово $\mathfrak{b}$ содержит подслово $(\exists x_i)$ или подслово $(\forall x_i)$. Если предметный символ x_i, встречающийся в слове $\mathfrak{b}$, не связан в этом слове, то он называется свободным. В частности, в первичных формулах все предметные символы свободны. Определяемые ниже «значения» формул не зависят от значений связанных переменных, а зависят лишь от значений свободных предметных символов и значений функциональных и предикатных символов.

б) Если $\mathfrak{A}$, $\mathfrak{B}$ — формулы, причем ни один из связанных предметных символов любой из этих формул не встре-

чается в другой, то формулами являются и слова

$$(\mathfrak{A} \,\&\, \mathfrak{B}),\ (\mathfrak{A} \vee \mathfrak{B}),\ (\mathfrak{A} \rightarrow \mathfrak{B}),\ \neg \mathfrak{A}.$$

Чтобы найти значение какой-либо из этих более сложных формул, надо найти значения формул $\mathfrak{A}$, $\mathfrak{B}$ и затем над этими значениями (равными *И* или *Л*) произвести соответствующую операцию &, $\vee$, $\rightarrow$, $\neg$.

в) Если $\mathfrak{A}$ — формула, содержащая свободный предметный символ x_i, то формулами являются и слова

$$(\forall x_i)\,\mathfrak{A},\ (\exists x_i)\,\mathfrak{A}.$$

Пусть задано множество M и на этом множестве заданы значения всех предикатных, функциональных и свободных предметных символов, входящих в формулу $\mathfrak{A}$. Тогда в формуле $\mathfrak{A}$ будет не задано значение лишь одного свободного предметного символа x_i. Для каждого значения x_i в M и фиксированных значений остальных символов значение формулы $\mathfrak{A}$ будет равно *И* или *Л*. Если $\mathfrak{A}$ для любых значений x_i в M имеет значение *И*, то говорят, что формула $(\forall x_i)\,\mathfrak{A}$ имеет значение *И* при фиксированных значениях входящих в нее свободных символов. Если же формула $\mathfrak{A}$ хотя бы при одном значении x_i из M имеет значение *Л*, то формула $(\forall x_i)\,\mathfrak{A}$ также имеет значение *Л*.

Аналогично, если при фиксированных значениях функциональных, предикатных и свободных предметных символов формулы $(\exists x_i)\,\mathfrak{A}$ формула $\mathfrak{A}$ ложна для любых значений предметного символа x_i в множестве M, то говорят, что формула $(\exists x_i)\,\mathfrak{A}$ имеет значение *Л* для фиксированных значений символов. В противном случае говорят, что $(\exists x_i)\,\mathfrak{A}$ имеет значение *И* на M.

Слово $\mathfrak{A}$ называется формулой, если оно является формулой в силу предписаний а), б), в).

При практическом использовании формул удобно употреблять сокращенную запись формул. Например, формулу $\neg(\mathfrak{a} = \mathfrak{b})$ обычно записывают в виде $\mathfrak{a} \neq \mathfrak{b}$. Внешние скобки обычно опускают; если T — двуместный предикатный или функциональный символ, то вместо $T\,(x,\ y)$ пишут xTy и т. д. Часто вместо самих символов x_i, P_n^m, F_n^m пишут их обозначения. Говорят, например, «Пусть P — двуместный предикатный символ, f — двуместный

функциональный символ, x, y—предметные переменные. Рассмотрим формулу

$$P(x, f(y, x)) \& P(x, x)». \tag{1}$$

На самом деле при этом имеют в виду формулу, которая получится из (1) после замены символов P, f, y, x соответствующими словами в алфавите $\boldsymbol{A}$.

Пусть $P_1, \ldots, P_s, F_1, \ldots, F_t$ — обозначения предикатных и функциональных символов, имеющих заданные числа мест, и $y_1, \ldots, y_u$ — обозначения предметных символов. Задать модель сигнатуры $\sigma = (P_1, \ldots, P_s, F_1, \ldots \ldots, F_t, y_1, \ldots, y_u)$ — значит задать некоторое непустое множество $\boldsymbol{M}$ (основное множество элементов модели) и задать на $\boldsymbol{M}$ значения всех указанных сигнатурных символов. Полученная модель обозначается через $\langle \boldsymbol{M}; P_1, \ldots \ldots, P_s, F_1, \ldots, F_t, y_1, \ldots, y_u \rangle$. Например, $\langle \boldsymbol{N}; + \rangle$, где $+$ есть символ бинарной операции, имеющий в качестве значения операцию сложения натуральных чисел, а $\boldsymbol{N}$—совокупность всех натуральных чисел, есть аддитивная полугруппа натуральных чисел. В этой полугруппе формулы

$$(\forall x)(\forall y)(x+y=y+x), (\forall x)(\forall y)(\exists z)(x+z=y \vee y+z=x)$$

истинны, а формула

$$(\exists z)(x+z=y)$$

определяет формульный предикат, совпадающий с обычным отношением $x \leqslant y$.

Формула $\mathfrak{A}$ называется формулой сигнатуры σ, если все ее предикатные, функциональные и свободные предметные символы (за исключением знака $=$) содержатся в σ. Поэтому, если задана какая-то модель $\mathfrak{M}$ сигнатуры σ, то каждая формула сигнатуры σ будет на этой модели истинной или ложной. Классом моделей сигнатуры σ называется произвольная система моделей заданной сигнатуры. В частности, класс может состоять из одной конкретной модели, а может состоять и из «всех» моделей данной сигнатуры.

Формула $\mathfrak{A}$ сигнатуры σ называется *истинной* на некотором классе моделей сигнатуры σ, если она истинна на каждой модели этого класса. Совокупность $T(\mathfrak{K})$ всех формул сигнатуры σ, истинных на классе $\mathfrak{K}$, называется

согласно А. Тарскому [97] *элементарной теорией* класса $\mathfrak{K}$.

Произвольная формула рассматриваемого исчисления называется *тождественно истинной*, если она истинна для любых значений входящих в нее функциональных, предикатных и свободных предметных символов. В частности, элементарная теория класса «всех» моделей данной сигнатуры σ есть совокупность всех тождественно истинных формул, имеющих сигнатуру σ.

Рассмотрим произвольную систему Σ каких-то формул заданной сигнатуры σ. Совокупность $K(\Sigma)$ всех моделей сигнатуры σ, на которых истинна каждая формула системы Σ, называется классом моделей, определенным системой аксиом Σ. Система Σ называется *выполнимой*, если класс $K(\Sigma)$ не пуст. В противном случае система Σ называется невыполнимой.

Например, пусть $\cdot$ служит обозначением некоторой бинарной операции. Тогда аксиома

$$(\forall x_1)(\forall x_2)(\forall x_3)((x_1 \cdot x_2) \cdot x_3 = x_1 \cdot (x_2 \cdot x_3)) \tag{2}$$

выражает ассоциативность операции $\cdot$ и класс моделей, определяемый указанной аксиомой, есть класс полугрупп с операцией $\cdot$.

Каждая аксиома системы Σ и каждая формула теории $T(K(\Sigma))$ являются словами в конечном алфавите $\boldsymbol{A}$. Поэтому имеет смысл ставить вопрос об алгоритмической природе совокупностей Σ, $T(K(\Sigma))$: будут ли они рекурсивными, креативными и т. п. Нижеследующие две фундаментальные теоремы дают ответ на этот вопрос в основных случаях.

Теорема 1 (Гильберт — Гедель). *Если Σ — рекурсивно перечислимая система формул заданной сигнатуры σ, то элементарная теория $T(K(\Sigma))$ также рекурсивно перечислима.*

Для доказательства сначала рассматривается совокупность $\mathfrak{T}$ вообще всех тождественно истинных формул (любой сигнатуры). В курсах математической логики (см., например, [17]) показывается, что формулы $\mathfrak{T}$ могут быть получены следующим способом. Выделяется некоторый класс тождественно истинных формул, называемых аксиомами исчисления. В разных курсах этот класс выбирается по-разному. Однако всегда оказывается вполне очевидным,

что аксиомы образуют рекурсивное множество. Затем указывается конечное число особых «правил вывода». Одно из них (правило силлогизма) имеет вид: если даны формулы $\mathfrak{A}$ и $\mathfrak{A} \to \mathfrak{B}$, то образуем формулу $\mathfrak{B}$. По существу эти правила являются просто словарными функциями, притом рекурсивными в алфавите A. Например, правилу силлогизма соответствует такая функция:

$$Q(\mathfrak{A}, Y) = \begin{cases} \mathfrak{B}, \text{ если } Y \text{ имеет вид } \mathfrak{A} \to \mathfrak{B}, \\ \mathfrak{A} \text{ в остальных случаях.} \end{cases}$$

Затем показывается, что все тождественно истинные формулы получаются с помощью операций вывода из аксиом, то есть что совокупность всех тождественно истинных формул *порождается* множеством аксиом с помощью операций вывода. Так как множество аксиом рекурсивно, а операции вывода являются рекурсивными функциями, то согласно теореме о порожденных множествах из п. 4.3 совокупность $\mathfrak{T}$ всех тождественно истинных формул рекурсивно перечислима.

Чтобы доказать теорему 1 в общем виде, обозначим через $\mathfrak{F}_0, \mathfrak{F}_1, \ldots$ рекурсивную последовательность всех формул из Σ. Известно, что формула $\mathfrak{A}$ тогда и только тогда принадлежит $T(K(\Sigma))$, когда для некоторого n формула

$$\mathfrak{F}_0 \,\&\, \mathfrak{F}_1 \,\&\, \ldots \,\&\, \mathfrak{F}_n \to \mathfrak{A} \qquad (3)$$

является тождественно истинной. Поэтому мы составляем множество $\mathfrak{M}$ всех формул вида (3), заставляя n пробегать натуральный ряд, а $\mathfrak{A}$ пробегать совокупность всех формул сигнатуры σ. Множество $\mathfrak{M}$ рекурсивно перечислимо. Поэтому рекурсивно перечислимо и его пересечение $\mathfrak{M}_t$ с множеством всех тождественно истинных формул. Пусть $\mathfrak{B}_0$, $\mathfrak{B}_1, \ldots$ — формулы этого пересечения. $\mathfrak{B}_i$ имеет форму

$$\mathfrak{F}_0 \,\&\, \mathfrak{F}_1 \,\&\, \ldots \,\&\, \mathfrak{F}_{n_i} \to \mathfrak{A}$$

и потому $T(K(\Sigma))$ будет состоять из членов рекурсивно перечислимой последовательности $\mathfrak{A}_0, \mathfrak{A}_1, \ldots, \mathfrak{A}_n, \ldots$, что и требовалось.

Теорема 2 (Чёрч [115]). *Совокупность $\mathfrak{T}$ всех тождественно истинных формул исчисления предикатов является креативным (и потому нерекурсивным) множеством.*

Для краткости обозначим точкой бинарный функциональный символ F_0^2 и рассмотрим класс $\mathfrak{P}$ всех полугрупп, определяемый аксиомой (2). Согласно теореме Поста — Маркова (п. 13.1) существует полугрупповое исчисление с порождающими элементами $x_1, \ldots, x_n$ и определяющими соотношениями

$$\mathfrak{a}_1 = \mathfrak{b}_1, \ \mathfrak{a}_2 = \mathfrak{b}_2, \ \ldots, \ \mathfrak{a}_s = \mathfrak{b}_s \qquad (4)$$

такое, что множество U всех слов, эквивалентных некоторому подходящему слову $\mathfrak{a}$, является креативным. Каждое слово $x_{i_1}x_{i_2} \ldots x_{i_m}$ этого полугруппового исчисления мы будем рассматривать как терм $((x_{i_1} \cdot x_{i_2}) \cdot x_{i_3}) \ldots x_{i_m}$.

Из определения эквивалентности слов в $\mathfrak{P}$ следует, что слово $\mathfrak{a}$ эквивалентно слову $\mathfrak{x}$ тогда и только тогда, когда для любых значений $x_1, \ldots, x_n$, взятых в произвольной полугруппе и удовлетворяющих в этой полугруппе соотношениям (4), значения термов $\mathfrak{a}$, $\mathfrak{x}$ будут равны, то есть когда формула

$$(\forall x_{n+1})(\forall x_{n+2})(\forall x_{n+3})(x_{n+1} \cdot (x_{n+2} \cdot x_{n+3}) = \\ = (x_{n+1} \cdot x_{n+2}) \cdot x_{n+3}) \,\&\, (\mathfrak{a}_1 = \mathfrak{b}_1 \,\&\, \ldots \,\&\, \mathfrak{a}_s = \mathfrak{b}_s) \longrightarrow \mathfrak{a} = \mathfrak{x} \qquad (5)$$

тождественно истинна. Итак, для любого слова $\mathfrak{x}$

$$\mathfrak{x} \in U \Longleftrightarrow F(\mathfrak{x}) \in \mathfrak{T},$$

где через $F(\mathfrak{x})$ обозначено слово (5). Ясно, что словарная функция $F(\mathfrak{x})$ рекурсивна. Таким образом, креативное множество U m-сводится к рекурсивно перечислимому множеству $\mathfrak{T}$ и потому (п. 8.2) множество $\mathfrak{T}$ креативно.

После того как на рубеже прошлого и нашего столетий был окончательно описан язык исчисления предикатов, естественно возникла задача о нахождении всех логических законов, формулируемых на этом языке, то есть о нахождении всех тождественно истинных формул, и о нахождении алгоритма, позволяющего для каждой данной формулы узнать, является ли она тождественно истинной. Последняя задача оставалась открытой в течение ряда лет и получила название проблемы разрешимости (Entscheidungsproblem) исчисления предикатов. В 1936 г. А. Чёрч [115] показал, что если принять тезис, носящий ныне его имя (п. 2.3), то проблема разрешимости решается отрицательно. Это был первый крупный успех только что родившейся в те годы теории алгоритмов. Теорема 2 представляет собой модернизированную версию первоначальной теоремы Чёрча, так как понятие креативного множества было введено Э. Постом много позже.

13.3. Арифметические множества. Рассмотрим натуральный ряд N, на котором отметим операции сложения $+$, умножения $\times$ и числа 0, 1. Получившуюся модель

$$\mathfrak{A} = \langle N;\ +,\ \times,\ 0,\ 1\rangle$$

назовем арифметикой. Символы $+$, $\times$, 0, 1 мы будем рассматривать как бинарные функциональные и предметные символы, значения которых фиксированы. Формула $\mathfrak{a}$ исчисления предикатов называется арифметической, если она не содержит предикатных символов, каждый ее функциональный символ есть либо $+$, либо $\times$, предметные символы 0, 1 в ней не связаны. Предметные символы 0, 1, имеющие фиксированные значения, называются *индивидуальными*. Остальные предметные символы, входящие в формулу $\mathfrak{a}$, называются *предметными переменными*. Арифметическая формула, не имеющая свободных предметных переменных, называется *замкнутой*. Каждая замкнутая арифметическая формула имеет на натуральном ряде значение *И* или *Л*. Те из них, которые имеют значение *И*, могут быть рассматриваемы как выражающие на языке исчисления предикатов свойства натурального ряда.

Пусть $\mathfrak{a}$ — арифметическая формула, свободные предметные переменные которой суть $y_1, \ldots, y_n$. Говорят, что формула $\mathfrak{a}$ истинна (или тождественно истинна) на $\mathfrak{A}$, если $\mathfrak{a}$ истинна для каждых значений $y_1, \ldots, y_n$ в N. Отсюда ясно, что формула $\mathfrak{a}$ со свободными предметными переменными $y_1, \ldots, y_n$ истинна на $\mathfrak{A}$ тогда и только тогда, когда на $\mathfrak{A}$ истинна замкнутая формула $(\forall y_1) \ldots \ldots (\forall y_n)\, \mathfrak{a}$. В общем случае значение формулы будет зависеть от значений переменных $y_1, \ldots, y_n$ и потому значение $\mathfrak{a}$ можно рассматривать как значения n-местного предиката $P(y_1, \ldots, y_n)$, определяемого или «изображаемого» формулой $\mathfrak{a}$. Предикаты, которые определяются арифметическими формулами, называются *арифметическими* (или формульными) *предикатами*. Например, предикаты $x < y$ и $x \mid y$ (x делит y) арифметические, так как

$$x < y \Leftrightarrow x \neq y \,\&\, (\exists z)(x + z = y),$$

$$x \mid y \Leftrightarrow (\exists z)(y = x \times z).$$

Множество n-ок натуральных чисел $\langle y_1, \ldots, y_n \rangle$ называется *арифметическим* (по Гёделю), если арифметическим является n-местный предикат, истинный на n-ках этого множества и ложный на остальных n-ках. Числовая частичная функция $y = f(x_1, \ldots, x_n)$ называется *арифметической*, если график ее есть множество арифметическое.

Для краткости терм вида $1 + 1 + \ldots + 1$ (a единиц в арифметических формулах обозначается числом a. Говорят, что множество M или функция f изображаются формулой $\mathfrak{a}$, если формулой $\mathfrak{a}$ изображается предикат, отвечающий M или f. Например, формула $y = 2$ представляет множество, состоящее из одного числа 2; формула $(\exists z)\ (y = 2 \times z)$ представляет совокупность всех четных чисел; формула

$$y \neq 0 \mathbin{\&} y \neq 1 \mathbin{\&} (\forall u)(\forall v)(y = uv \rightarrow u = 1 \vee v = 1)$$

представляет совокупность всех простых чисел; формула

$$(\exists u)((z \times z) + u = y) \mathbin{\&}$$
$$\mathbin{\&} (\exists v)((z+1) \times (z+1) = y + v) \mathbin{\&} y \neq (z+1) \times (z+1)$$

представляет функцию $z = [\sqrt{y}]$ и т. д.

Теорема 1 (Гедель). *Каждая частично рекурсивная функция, а потому и каждое рекурсивно перечислимое множество n-ок являются арифметическими.*

Согласно следствию из теоремы 3 в п. 3.4 и теореме о нормальной форме Клини (п. 6.1) каждая частично рекурсивная функция может быть получена из функций $\mathbf{\Gamma}$, $\boldsymbol{c}$, $\boldsymbol{l}$, $\boldsymbol{r}$, $x + y$, $x \dot{-} y$, $x + 1$, I_n^m операциями подстановки и минимизации. Поэтому теорема 1 будет доказана, если мы докажем, что все указанные исходные функции арифметические и что операции подстановки и минимизации, примененные к арифметическим функциям, дают функции арифметические.

А) Суперпозиция

$$f(x_1, \ldots, x_m) = g(g_1(x_1, \ldots, x_m), \ldots, g_n(x_1, \ldots, x_m))$$

арифметических функций $g, g_1, \ldots, g_n$ есть функция арифметическая.

Действительно, если формулы $G(x_1, \ldots, x_n, y)$ и $G_i(x_1, \ldots, x_m, y)$ представляют функции $\boldsymbol{g}$, $\boldsymbol{g}_i$, то

формула

$$(\exists z_1) \ldots (\exists z_n)(G(z_1, \ldots, z_n, y) \,\&\, G_1(x_1, \ldots, x_m, z_1) \,\&\, \ldots$$
$$\ldots \,\&\, G_n(x_1, \ldots, x_m, z_n))$$

представляет, очевидно, функцию $y = f(x_1, \ldots, x_n)$.

Б) Если формула $G(x_1, \ldots, x_{n+1}, u)$ представляет частичную функцию $g(x_1, \ldots, x_{n+1}) = u$, то формула

$$G(x_1, \ldots, x_n, y, 0) \,\&\, (\forall z)(z < y \rightarrow$$
$$\rightarrow (\exists u)(G(x_1, \ldots, x_n, z, u) \,\&\, u \neq 0))$$

представляет частичную функцию

$$y = \mu_z(g(x_1, \ldots, x_n, z) = 0).$$

Таким образом, оператор минимизации, примененный к арифметической функции, дает функцию арифметическую.

В) Функции $+$, $x + 1$, $x \dot{-} y$, I_n^m изображаются соответственно формулами

$$z = x + y, \quad z = x + 1, \quad (y < x \rightarrow y + z = x) \,\&\, (x \leqslant y \rightarrow z = 0),$$
$$z = 0 \cdot x_1 + \ldots + x_m + \ldots + 0 \cdot x_n$$

и потому являются арифметическими.

Г) Функции x^2, $[x/2]$, $c(x, y)$, $\boldsymbol{l}(x)$, $\boldsymbol{r}(x)$ изображаются формулами

$$z = x \times x, \quad (2 \times z \leqslant x) \,\&\, (x < 2 \times (z + 1)),$$
$$\left[\frac{(x+y)^2 + 3 \times x + y}{2}\right], \quad (\exists y)(x = c(z, y)), \quad (\exists y)(x = c(z, y))$$

и потому являются арифметическими.

Д) Функции $z = \text{rest}(x, y)$, $z = \boldsymbol{\Gamma}(x, y)$ изображаются соответственно формулами

$$\exists u(x = uy + z \,\&\, z < y) \vee (y = 0 \,\&\, z = x),$$
$$z = \text{rest}(\boldsymbol{l}(x), 1 + (y + 1)\boldsymbol{r}(x)).$$

Согласно приведенному выше замечанию из А) — Д) непосредственно вытекает теорема 1.

Каждая арифметическая формула может рассматриваться как слово в конечном алфавите $J = \{\&, \vee, \neg, \forall, \exists, (,), ', +, \times, =, 0, 1, x, \alpha\}$, причем слова $x_i = (x\alpha \ldots \alpha)$ можно употреблять в качестве кодов для предметных симво-

лов. Все слова в алфавите J, а значит, и все арифметические формулы, имеют определенные алфавитные номера. Множество арифметических формул называется арифметическим, если арифметическим является множество номеров этих формул.

Следствие. *Множество всех замкнутых истинных арифметических формул не рекурсивно и не рекурсивно перечислимо.*

Пусть $f(x)$ — примитивно рекурсивная функция, совокупность значений которой U не рекурсивна. Согласно теореме 1 существует арифметическая формула $\mathfrak{a}(x, y)$, представляющая функцию $y = f(x)$, и, следовательно,

$$y \in U \Leftrightarrow (\exists x)\, \mathfrak{a}(x, y) \quad (y = 0, 1, 2, \ldots).$$

Таким образом, если бы существовал алгоритм, распознающий истинные замкнутые формулы, то при помощи этого алгоритма можно было бы распознавать числа, принадлежащие U, а это противоречит нерекурсивности U.

Допустим теперь, что совокупность V всех истинных замкнутых арифметических формул рекурсивно перечислима. Пусть

$$V_0, V_1, \ldots, V_n, \ldots$$

— рекурсивная последовательность всех формул из V. Тогда рекурсивная последовательность

$$\neg V_0, \neg V_1, \ldots, \neg V_n, \ldots$$

была бы последовательностью всех *ложных* замкнутых арифметических формул, то есть совокупность W всех ложных замкнутых арифметических формул была бы рекурсивно перечислимой. Ясно, что совокупность всех замкнутых арифметических формул рекурсивна. Эта совокупность разложена в сумму непересекающихся рекурсивно перечислимых множеств V и W. По теореме Поста отсюда следует, что V и W рекурсивны, вопреки уже доказанной нерекурсивности множества V.

Согласно теореме 1 все рекурсивно перечислимые множества арифметические. Обратное, конечно, неверно. Пусть множество M определяется арифметической формулой $\mathfrak{P}$. Тогда формула $\neg\mathfrak{P}$ будет определять дополнение $M' = N - M$. Поэтому дополнения к рекурсивно перечисли-

мым множествам являются также арифметическими множествами. Можно легко построить и более сложные арифметические множества, не являющиеся ни рекурсивно перечислимыми, ни дополнениями к последним.

Теорема 2 (А. Тарский). *Совокупность номеров всех истинных (замкнутых) арифметических формул не является арифметической.*

Для доказательства мы используем обычный метод универсальных функций. А именно, предполагая теорему *ложной*, будем стремиться построить арифметическую формулу $\mathfrak{P}(x, y)$ с двумя свободными предметными переменными x, y, обладающую следующим свойством универсальности: если a — номер некоторой арифметической формулы $\mathfrak{Q}(y)$, имеющей лишь одну свободную предметную переменную y, то для всех значений y

$$\mathfrak{P}(a, y) \Leftrightarrow \mathfrak{Q}(y). \quad (1)$$

Допустим, что такую формулу $\mathfrak{P}$ нам удалось найти. Пусть a — номер формулы $\neg \mathfrak{P}(y, y)$. Из (1) получаем, что

$$\mathfrak{P}(a, y) \Leftrightarrow \neg \mathfrak{P}(y, y)$$

для произвольного значения y. Беря для y значение a, получаем противоречивое соотношение

$$\mathfrak{P}(a, a) \Leftrightarrow \neg \mathfrak{P}(a, a),$$

которое и доказывает теорему.

Итак, пусть теорема ложна и, следовательно, существует арифметическая формула $\mathfrak{B}(y)$ с единственной свободной переменной y такая, что для каждого натурального n $\mathfrak{B}(\tilde{n}) = И$ *) тогда и только тогда, когда n есть номер слова, являющегося истинной замкнутой арифметической формулой.

Для каждого натурального n через $\mathfrak{a}_n(i)$ обозначим слово, получающееся следующим образом: если n есть номер слова $\mathfrak{a}_n$, являющегося арифметической формулой, содержащей единственную свободную предметную переменную y, то полагаем

$$\mathfrak{a}_n'(i) = \mathrm{Sb}(\mathfrak{a}_n;\ y,\ \tilde{i});$$

*) $\tilde{n}$ обозначает терм $1 + \ldots + 1$.

если же n есть номер слова $\mathfrak{a}_n$, не удовлетворяющего указанным требованиям, то полагаем $\mathfrak{a}_n(i) = \mathfrak{a}_n$.

Номер слова $\mathfrak{a}_n(i)$ обозначим через $f(n, i)$. Таким образом, если $\mathfrak{a}_n$ есть формула с единственной свободной предметной переменной y, то для любого натурального i

$$\mathfrak{B}(f(n, i)) = И \Leftrightarrow \mathfrak{a}_n(i) = И. \qquad (2)$$

На основе результатов п. 11.2 легко показать, что функция $f(n, i)$ рекурсивна. Поэтому существует арифметическая формула $\mathfrak{F}(x, y, u)$ с тремя свободными предметными переменными x, y, u, представляющая отношение $u = f(x, y)$. Положим

$$\mathfrak{P}(x, y) \overset{df}{=} (\exists u)(\mathfrak{F}(x, y, u) \,\&\, \mathfrak{B}(u))$$

и покажем, что формула $\mathfrak{P}(x, y)$ удовлетворяет требованию (1). В самом деле, пусть n — номер формулы $\mathfrak{Q}(y)$. Тогда для любого натурального i формула $\mathfrak{Q}(\tilde{i})$ совпадает с формулой $\mathfrak{a}_n(i)$ и потому в силу (2)

$$\mathfrak{B}(\widetilde{f}(n, i)) \Leftrightarrow \mathfrak{Q}(\tilde{i}). \qquad (3)$$

Пусть $\mathfrak{Q}(\tilde{i}) = И$. Тогда $\mathfrak{B}(\tilde{a}) = И$, где положено $a = f(n, i)$. Из последнего равенства заключаем, что $\mathfrak{F}(\tilde{n}, \tilde{i}, \tilde{a}) = И$, $\mathfrak{F}(\tilde{n}, \tilde{i}, \tilde{a}) \,\&\, \mathfrak{B}(\tilde{a}) = И$ и потому

$$\mathfrak{P}(\tilde{n}, \tilde{i}) = (\exists u)(\mathfrak{F}(\tilde{n}, \tilde{i}, u) \,\&\, \mathfrak{B}(u)) = И. \qquad (4)$$

Обратно, пусть для фиксированных n, i соотношение (4) истинно. Это означает, что существует такое натуральное число a, для которого

$$\mathfrak{F}(\tilde{n}, \tilde{i}, \tilde{a}) \,\&\, \mathfrak{B}(\tilde{a}) = И.$$

Из $\mathfrak{F}(\tilde{n}, \tilde{i}, \tilde{a}) = И$ следует, что $a = f(n, i)$ и, значит,

$$\mathfrak{B}(\widetilde{f}(n, i)) = \mathfrak{B}(\tilde{a}) = И.$$

Из (3) получаем, что $\mathfrak{Q}(\tilde{i}) = И$. Итак, эквивалентность (1) доказана, а вместе с нею доказана и теорема 2.

13.4. Формулы 2-й ступени. В рассмотренных выше предикатных формулах все предикатные и функциональные символы были свободными. Если правила образования формул а), б), в) из п. 13.2 дополнить еще правилом

г) если X — предикатный или функциональный символ, свободный в формуле $\mathfrak{A}$, то формулами будут и слова

$$(\forall X)\,\mathfrak{A}, \quad (\exists X)\,\mathfrak{A},$$

то получающийся более широкий класс формул называется классом формул 2-й ступени. «Значение» формулы 2-й ступени при заданных значениях свободных предикатных, функциональных и предметных символов определяется так же, как и для формул 1-й ступени.

Формулы 2-й ступени, не содержащие ни функциональных, ни предикатных, ни предметных свободных символов, называются абсолютно замкнутыми. На любом непустом множестве M абсолютно замкнутая формула либо истинна, либо ложна, причем значение этой формулы на M зависит лишь от числа элементов (мощности) M. Формула 2-й ступени (не обязательно абсолютно замкнутая) называется тождественно истинной, если она истинна на любом непустом множестве M при любых значениях на M всех ее свободных символов.

Теорема 1. *Множество всех тождественно истинных формул второй ступени не рекурсивно перечислимо.*

Пусть $'$ — одноместный, $+$, $\times$ — двуместные функциональные символы, P — одноместный предикатный символ, 0, 1, x, y — предметные символы. Рассмотрим следующую систему формул:

1. $(\forall x)\,(x' \neq 0 \,\&\, 0' = 1)$,
2. $(\forall x)\,(\forall y)\,(x' = y' \rightarrow x = y)$,
3. $(\forall P)\,(P(0) \,\&\, (\forall x)\,(P(x) \rightarrow P(x')) \rightarrow (\forall y)\,P(y))$,
4. $(\forall x)\,(x + 0 = x)$,
5. $(\forall x)\,(\forall y)\,(x + y' = (x + y)')$,
6. $(\forall x)\,(x \times 0 = 0)$,
7. $(\forall x)\,(\forall y)\,(x \times y' = (x \times y) + x)$.

Смысл этих формул очевиден. Например, формула 3) является принципом полной индукции для натурального ряда. Обозначим через $\mathfrak{S}$ конъюнкцию формул 1) — 7). Формула $\mathfrak{S}$ содержит свободные символы $'$, $+$, $\times$, 0, 1. Известно, что если на некотором множестве M значения этих символов заданы так, что формула $\mathfrak{S}$ истинна, то модель $\langle M;\, +,\, \times,\, 0,\, 1\rangle$ изоморфна обычной арифметике

$\langle N; +, \times, 0, 1\rangle$, рассмотренной в п. 13.3. Поэтому для каждой замкнутой арифметической (в смысле п. 13.3) формулы $\mathfrak{x}$ формула 2-й ступени

$$\mathfrak{S} \to \mathfrak{x}$$

тождественно истинна тогда и только тогда, когда $\mathfrak{x}$ истинна в арифметике $\langle N; +, \times, 0, 1\rangle$.

Ясно, что совокупность всех замкнутых арифметических формул $\mathfrak{x}$ (истинных и ложных) рекурсивна. Поэтому рекурсивна и совокупность $\mathfrak{M}$ всех формул вида $\mathfrak{S} \to \mathfrak{x}$. Если бы совокупность $\mathfrak{T}$ всех тождественно истинных формул 2-й ступени была рекурсивно перечислимой, то пересечение $\mathfrak{M} \cap \mathfrak{T}$ было бы также рекурсивно перечислимой совокупностью и ее формулы можно было бы выписать в рекурсивную последовательность

$$\mathfrak{S} \to \mathfrak{x}_0, \ \mathfrak{S} \to \mathfrak{x}_1, \ \ldots, \ \mathfrak{S} \to \mathfrak{x}_n, \ \ldots$$

Но тогда рекурсивная последовательность $\mathfrak{x}_0, \mathfrak{x}_1, \ldots$ состояла бы из всех замкнутых арифметических истинных формул и, следовательно, совокупность всех последних формул была бы рекурсивно перечислимой, вопреки основному результату п. 13.3.

Теорема 1 данного п. 13.4 и теорема 1 из п. 13.2 выявляют глубокое различие между языками 1-й и 2-й ступени. Хотя множество законов, формулируемых на языке 1-й ступени, нерекурсивно, все же существует алгоритм, позволяющий постепенно строить все эти законы. Что касается логических законов, формулируемых на языке 2-й ступени, то совокупность их не только не рекурсивна, но и не перечислима.

Дополнения и примеры

1. Пусть $\mathfrak{C}$ — ассоциативное исчисление с порождающими элементами $c_1, \ldots, c_s$ и определяющими соотношениями $\mathfrak{a}_\alpha \equiv \mathfrak{b}_\alpha$ $(\alpha = 1, \ldots, l)$. Для произвольного слова $\mathfrak{x}$ в алфавите $c_1, \ldots, c_s$ через $\mathfrak{x}^*$ обозначим слово в алфавите a, b, получающееся из $\mathfrak{x}$ заменой $c_i = aba^{i+1}b^{i+1}$ $(i = 1, \ldots, s)$. Обозначим через $\mathfrak{C}^*$ ассоциативное исчисление с порождающими элементами a, b и определяющими соотношениями $\mathfrak{a}_\alpha^* \equiv \mathfrak{b}_\alpha^*$ $(\alpha = 1, \ldots, l)$. Показать, что для произвольных слов $\mathfrak{a}, \mathfrak{b}$ в алфавите $c_1, \ldots, c_s$ $\mathfrak{a} \equiv \mathfrak{b}$ в исчислении $\mathfrak{C}$ тогда и только тогда, когда $\mathfrak{a}^* \equiv \mathfrak{b}^*$ в исчислении $\mathfrak{C}^*$.

В частности, если в $\mathfrak{C}$ проблема равенства неразрешима, то она неразрешима в исчислении $\mathfrak{C}^*$ (Холл [111]).

2. Говорят, что в ассоциативном исчислении $\mathfrak{C}$ разрешима проблема *делимости слева*, если существует алгоритм, позволяющий для любых двух слов $\mathfrak{a}$, $\mathfrak{b}$ из $\mathfrak{C}$ сказать, существует ли решение $\mathfrak{x}$ уравнения $\mathfrak{a}\mathfrak{x} \equiv \mathfrak{b}$ в $\mathfrak{C}$. Аналогично говорят, что в $\mathfrak{C}$ разрешима проблема делимости слева *заданного элемента* $\mathfrak{c}$, если существует алгоритм, позволяющий для любого слова $\mathfrak{a}$ из $\mathfrak{C}$ сказать, существует или нет решение $\mathfrak{x}$ уравнения $\mathfrak{a}\mathfrak{x} \equiv \mathfrak{c}$ в $\mathfrak{C}$. Показать, что в ассоциативном исчислении $A(\mathfrak{T})$, построенном в п. 13.1 в процессе доказательства теоремы 3, проблема делимости слева элемента $q_0 v$ не разрешима (ср. С. И. Адян [1]).

3. В предыдущем примере в ассоциативном исчислении $A(\mathfrak{T})$ рассматривались только непустые слова. Допуская в число слов и пустое слово, получаем ассоциативное исчисление с пустым словом. Так как $[\Lambda][\mathfrak{a}] = [\Lambda\mathfrak{a}] = [\mathfrak{a}]$ и $[\mathfrak{a}][\Lambda] = [\mathfrak{a}]$, то пустое слово Λ является представителем класса $[\Lambda]$, который служит единицей полугруппы, определяемой этим исчислением. Если среди определяющих соотношений исчисления находится соотношение вида $\mathfrak{a} \equiv \Lambda$, то правым элементарным преобразованием, отвечающим этому соотношению, называется преобразование, переводящее заданное слово $\mathfrak{x}\mathfrak{y}$ в одно из слов $\mathfrak{a}\mathfrak{x}\mathfrak{y}$, $\mathfrak{x}\mathfrak{a}\mathfrak{y}$, $\mathfrak{x}\mathfrak{y}\mathfrak{a}$. Левым элементарным преобразованием, отвечающим указанному соотношению, называется преобразование $\mathfrak{a} \to \Lambda$, переводящее слова вида $\mathfrak{a}\mathfrak{x}\mathfrak{y}$, $\mathfrak{x}\mathfrak{a}\mathfrak{y}$, $\mathfrak{x}\mathfrak{y}\mathfrak{a}$ в слово $\mathfrak{x}\mathfrak{y}$ ($\mathfrak{x}$, $\mathfrak{y}$ могут быть и пустыми словами).

Говорят, что в ассоциативном исчислении с пустым словом разрешима проблема нахождения правого обратного элемента, если совокупность тех слов $\mathfrak{a}$, для которых уравнение $\mathfrak{a}\mathfrak{y} \equiv \Lambda$ имеет решение в данном исчислении, является рекурсивной. Легко строится ассоциативное исчисление с пустым словом и неразрешимой проблемой нахождения правого обратного элемента. Для этого берем ассоциативное исчисление $A(\mathfrak{T})$, построенное в процессе доказательства теоремы 3 из п. 13.1. К алфавиту этого исчисления добавляем новую букву u, к определяющим соотношениям добавляем соотношение $uq_0v \equiv \Lambda$ и новое исчисление с пустым словом обозначаем через $B(\mathfrak{T})$. По аналогии с теоремой 2 из п. 13.1 показать, что в исчислении $B(\mathfrak{T})$ $u\mathfrak{a}q_w\mathfrak{b}v \equiv \Lambda$ тогда и только тогда, когда $\mathfrak{a}q_w\mathfrak{b}v \equiv q_0v$ в исчислении $A(\mathfrak{T})$. Отсюда путем сравнения с предыдущей задачей видим, что в $B(\mathfrak{T})$ не разрешима проблема нахождения правого обратного элемента.

4. Пусть $\mathfrak{T}$ — машина Тьюринга, правильно вычисляющая функцию $E(x)$, упомянутую в доказательстве теоремы 3 из п. 13.1. Обозначим через $\mathfrak{B}$ ассоциативное исчисление с порождающими элементами $0, 1, q_0, q_1, \ldots, q_n$ и соотношениями (5), (6), (7) из п. 13.1 (по сравнению с $A(\mathfrak{T})$ нет порождающего элемента v и соотношения (8)). Показать, что в $\mathfrak{B}$ проблема равенства разрешима, но неразрешима следующая проблема: по данным словам $\mathfrak{a}$, $\mathfrak{b}$ узнать, существует ли натуральное число x такое, что $\mathfrak{a}0^x \equiv \mathfrak{b}0^x$.

5. Пусть $f(x)$ — рекурсивная числовая функция и $\mathfrak{G}$ — групповое исчисление с порождающими элементами $a, a^{-1}, b, b^{-1}, c, c^{-1}, d, d^{-1}$ и (рекурсивно перечислимой) бесконечной совокупностью

определяющих соотношений

$$aa^{-1} \equiv bb^{-1} \equiv cc^{-1} \equiv dd^{-1} \equiv \Lambda,$$
$$b^{f(x)}a\,(b^{-1})^{f(x)} \equiv d^{f(x)}c\,(d^{-1})^{f(x)} \quad (x=0,\ 1,\ 2,\ \ldots).$$

Если совокупность значений функции f не рекурсивна, то проблема равенства слов в $\mathfrak{G}$ неразрешима (Рабин [79], Хигман [110]).

6. Пусть $\mathfrak{A}$ — групповое исчисление с порождающими элементами $a_1,\ a_1^{-1},\ \ldots,\ a_s,\ a_s^{-1}$ и некоторым бесконечным рекурсивно перечислимым множеством определяющих соотношений. Тогда существует групповое исчисление $\mathfrak{B}$ с порождающими элементами $a_1,\ a_1^{-1},\ \ldots,\ a_s,\ a_s^{-1}$ и некоторыми дополнительными порождающими элементами $b_1,\ b_1^{-1},\ \ldots,\ b_t,\ b_t^{-1}$, имеющее конечное число определяющих соотношений и такое, что эквивалентность произвольных двух слов в алфавите $a_1,\ a_1^{-1},\ \ldots,\ a_s,\ a_s^{-1}$ в группе $\mathfrak{A}$ равносильна эквивалентности этих слов в группе $\mathfrak{B}$ (то есть $\mathfrak{A}$ является подгруппой в группе $\mathfrak{B}$). Беря в качестве $\mathfrak{A}$ группу, построенную в предыдущей задаче, и конструируя для нее группу $\mathfrak{B}$, получим групповое исчисление с конечным числом определяющих соотношений и неразрешимой проблемой равенства (Хигман [110]).

7. Пусть $\mathfrak{A}$ — свободная полугруппа с свободными порождающими элементами $a,\ b$, $\mathfrak{A}^2$ — декартов квадрат $\mathfrak{A}$, то есть полугруппа пар $(\mathfrak{a},\ \mathfrak{b})$ $(\mathfrak{a},\ \mathfrak{b} \in \mathfrak{A})$, перемножаемых по закону

$$(\mathfrak{a},\ \mathfrak{b})(\mathfrak{c},\ \mathfrak{d}) = (\mathfrak{ac},\ \mathfrak{bd}).$$

Диагональная подполугруппа — это совокупность пар вида $(\mathfrak{a},\ \mathfrak{a})$. *Общая комбинаторная проблема Поста*: существует ли алгоритм, позволяющий для любой конечной системы пар $(\mathfrak{a}_1,\ \mathfrak{b}_1)$, $(\mathfrak{a}_2,\ \mathfrak{b}_2),\ \ldots,\ (\mathfrak{a}_t,\ \mathfrak{b}_t)$ решить, пусто или нет пересечение диагональной подполугруппы и подполугруппы, порожденной заданной системой пар.

Ограниченная комбинаторная проблема Поста: для фиксированного натурального числа $s > 0$ указать алгоритм, позволяющий для любой системы, состоящей из s пар $(\mathfrak{a}_1,\ \mathfrak{b}_1)$, $(\mathfrak{a}_2,\ \mathfrak{b}_2),\ \ldots\ \ldots,\ (\mathfrak{a}_s,\ \mathfrak{b}_s)$ решить, пусто или нет пересечение диагональной подполугруппы и подполугруппы, порожденной упомянутыми s парами.

Показать, что ограниченная проблема Поста (для больших s) решается отрицательно (Пост [72], Марков [55]).

8. В полугруппе всех целочисленных матриц 4-го порядка существует конечное число матриц таких, что порождаемая ими подполугруппа нерекурсивна (А. А. Марков [56]).

9. Прямое произведение двух свободных групп с двумя порождающими элементами содержит нерекурсивную подгруппу с конечным числом порождающих элементов. Отсюда следует, что группа целочисленных матриц 4-го порядка с определителем 1 содержит нерекурсивную подгруппу, имеющую конечное число порождающих элементов (К. А. Михайлова [61]).

10. В п. 13.2 доказано, что элементарная теория класса всех полугрупп нерекурсивна. В книге Тарского, Мостовского и Робинсона [97] и в обзоре Тайманова,

Тайцлина, Ершова и Лаврова [34] подробно исследованы элементарные теории многих важных классов моделей и алгебр. Некоторые из относящихся сюда результатов указаны в этой и следующих задачах.

Показать, что элементарная теория полугруппы $\langle N; +\rangle$ рекурсивна (Пресбургер, см. [34]).

11. Совокупность всех истинных абсолютно замкнутых формул 2-й ступени, не содержащих функциональных символов и содержащих из предикатных символов лишь знак равенства и одноместные предикатные символы, является рекурсивной.

12. Элементарные теории кольца всех действительных чисел $\langle C; +, \times\rangle$ и кольца всех комплексных чисел рекурсивны (А. Тарский [98]).

13. Элементарная теория кольца всех целых чисел нерекурсивна.

14. Элементарная теория поля всех рациональных чисел нерекурсивна (Ю. Робинсон).

15. Элементарные теории всех метабелевых групп и всех конечных метабелевых групп неразрешимы (А. И. Мальцев [50]).

16. Элементарные теории всех конечных симметрических групп и всех конечных простых групп неразрешимы (Ю. Л. Ершов [35]).

17. Говорят, что предикат $P(x_1, \ldots, x_n)$, определенный на множестве натуральных чисел N, принадлежит классу $\boldsymbol{P}_s$ иерархии Клини, если его можно представить в виде

$$P(\mathfrak{x}) \Longleftrightarrow \forall\mathfrak{y}_1\exists\mathfrak{y}_2 \cdots \forall\mathfrak{y}_{s-1}\exists\mathfrak{y}_s(F(\mathfrak{x}, \mathfrak{y}_1, \cdots, \mathfrak{y}_s)=0) \quad (s \text{ четно})$$

или соответственно в виде

$$P(\mathfrak{x}) \Longleftrightarrow \exists\mathfrak{y}_1\forall\mathfrak{y}_2 \ldots \forall\mathfrak{y}_{s-1}\exists\mathfrak{y}_s(F(\mathfrak{x}, \mathfrak{y}_1, \cdot\ , \mathfrak{y}_s)=0) \ (s \text{ нечетно}),$$

где $F(\mathfrak{x}, \mathfrak{y}_1, \ldots, \mathfrak{y}_s)$ — подходящая общерекурсивная функция. Предикат P называется принадлежащим классу $\boldsymbol{Q}_s$, если $\neg P$ принадлежит классу $\boldsymbol{P}_s$. Наконец, говорят, что предикат P принадлежит классу $\boldsymbol{R}_s$, если P принадлежит одновременно классу $\boldsymbol{P}_s$ и классу $\boldsymbol{Q}_s$. Отсюда следует, что класс $\boldsymbol{P}_1$ совпадает с классом всех рекурсивно перечислимых предикатов, а класс $\boldsymbol{R}_1$ — с классом рекурсивных предикатов. Далее,

$$\boldsymbol{P}_s \subset \boldsymbol{P}_{s+1}, \quad \boldsymbol{P}_s \subset \boldsymbol{Q}_{s+1}, \quad \boldsymbol{Q}_s \subset \boldsymbol{P}_{s+1}, \quad \boldsymbol{Q}_s \subset \boldsymbol{Q}_{s+1}.$$

При помощи нумерационных функций C^n, C_{ni} легко доказывается, что каждый предикат $P(\mathfrak{x})$ класса $\boldsymbol{P}_s$ представим в форме

$$P(\mathfrak{x}) \Longleftrightarrow \forall Y_1\exists Y_2 \ldots \forall Y_{s-1}\exists Y_s(F(\mathfrak{x}, Y_1, \ldots, Y_s)=0)$$

(F — общерекурсивная функция) для четных s и в аналогичной форме для нечетных s.

18. Объединение классов $\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots$ совпадает с классом всех арифметических предикатов.

19. Наряду с иерархией Клини иногда рассматривают и арифметическую иерархию. А именно, говорят, что формула вида

$$(Q_1\mathfrak{y}_1) \ldots (Q_s\mathfrak{y}_s)\mathfrak{A}(\mathfrak{x}, \mathfrak{y}_1, \ldots, \mathfrak{y}_s),$$

где символы Q_i означают кванторы $\forall$, $\exists$, $Q_i \neq Q_{i+1}$, $\mathfrak{y}_i = \langle y_{i1}, \ldots$ $\ldots, y_{im_i} \rangle$, $\mathfrak{A}$ — арифметическая формула, не содержащая кванторов и, следовательно, построенная из предметных символов $0, 1, x_1, \ldots, x_n, y_{ij}$, функциональных символов $+$, $\times$ и логических символов $=$, $\&$, $\vee$, $\neg$, имеет тип $Q_1 \ldots Q_s$. Например, формула

$$\forall x \forall y \exists u \forall v \, (x+y=0 \vee (x+y)(u+v) \neq 0)$$

имеет тип $\forall\exists\forall$. Определенный на натуральном ряде предикат $P(\mathfrak{x})$ называется предикатом типа $Q_1 \ldots Q_s$, если существует такая арифметическая формула $\mathfrak{A}(\mathfrak{x})$ данного типа, что $P(\mathfrak{x}) \Longleftrightarrow \mathfrak{A}(\mathfrak{x})$, где $\mathfrak{A}$ не содержит свободных предметных переменных, отличных от $x_1, \ldots, x_n$. Каждый арифметический предикат типа $(\forall\exists)^s$ принадлежит классу $\boldsymbol{P}_{2s}$ иерархии Клини и, аналогично, арифметические предикаты типов $\exists(\forall\exists)^s$, $(\forall\exists)^s\forall$, $(\exists\forall)^s$ принадлежат соответственно классам $\boldsymbol{P}_{2s+1}$, $\boldsymbol{Q}_{2s+1}$, $\boldsymbol{Q}_{2s}$.

20. В п. 16.2 показано, что каждый рекурсивный предикат имеет одновременно типы $\forall\exists$ и $\exists\forall$. Поэтому предикат класса $\boldsymbol{P}_{2s}$ заведомо имеет тип $(\forall\exists)^s\forall$.

21. Множество натуральных чисел $\boldsymbol{M}$ имеет данный класс в иерархии Клини, если этот класс имеет соответствующий одноместный предикат. Аналогично определяется и арифметический тип множества. Показать, что множество номеров *замкнутых* истинных арифметических формул типа $(\forall\exists)^s$ имеет класс $\boldsymbol{P}_{2s}$, типа $(\exists\forall)^s$ имеет класс $\boldsymbol{Q}_{2s}$ и т. д. Обратно, каждое множество класса $\boldsymbol{P}_{2s}$ имеет тип $(\forall\exists)^s\forall$ и т. д.

22. Легко видеть, что если бы для некоторого s все множества класса $\boldsymbol{P}_s$ принадлежали классу $\boldsymbol{Q}_s$ или все множества класса $\boldsymbol{Q}_s$ принадлежали классу $\boldsymbol{P}_s$, то все высшие классы $\boldsymbol{P}_{s+1}$, $\boldsymbol{P}_{s+2}$, ... совпадали бы и, следовательно (см. предыдущую задачу), совокупность номеров всех истинных арифметических формул была бы арифметической. Таким образом, для каждого s $\boldsymbol{P}_s \not\subseteq \boldsymbol{Q}_s$, $\boldsymbol{Q}_s \not\subseteq \boldsymbol{P}_s$ (теорема иерархии Клини) (см., например, Девис [23], стр. 156).

23. Пересечение и объединение конечного числа множеств, принадлежащих одному из классов $\boldsymbol{P}_s$, $\boldsymbol{Q}_s$, $\boldsymbol{R}_s$, принадлежит тому же классу. Класс множества не меняется при взаимно однозначных отображениях натурального ряда на себя. Если M_0 — множество четных чисел, имеющее класс $\boldsymbol{P}_s$ и не принадлежащее классу $\boldsymbol{Q}_s$, а M_1 — множество нечетных чисел, имеющее класс $\boldsymbol{Q}_s$ и не принадлежащее классу $\boldsymbol{P}_s$, то $M_0 \cup M_1$ принадлежит классу $\boldsymbol{R}_{s+1}$ и не принадлежит классам $\boldsymbol{P}_s$, $\boldsymbol{Q}_s$.

24. Если $P(x_1, \ldots, x_m)$ — предикат одного из классов Клини и $f(x_1, \ldots, x_n)$ — общерекурсивная функция, то предикат $P(f(x_1, \ldots, x_n), x_2, \ldots, x_m)$ принадлежит тому же классу, что и P.

25. Классом частичной функции называется класс ее графика. Класс предиката содержится в классе его характеристической

функции. Если предикат класса $\boldsymbol{P}_s$ или класса $\boldsymbol{Q}_s$, то характеристическая функция этого предиката лежит в классе $\boldsymbol{R}_{s+1}$.

26. Доказать формулы

$$(\exists x \leqslant z)\, \forall y P(x, y, u) \Longleftrightarrow \forall y\, (\exists x \leqslant z)\, P(x, \Gamma(y, x), u),$$

$$(\forall x \leqslant z)\, \exists y P(x, y, u) \Longleftrightarrow \exists y\, (\forall x \leqslant z)\, P(x, \Gamma(y, x), u),$$

где P — произвольный предикат, $\Gamma(x, y)$ — функция Геделя.

27. Суперпозиция всюду определенных функций класса $\boldsymbol{R}_s$ принадлежит тому же классу $\boldsymbol{R}_s$.

28. Если функция $f(x, y, z)$ принадлежит классу $\boldsymbol{R}_s$ и функция $g(x, y) = \mu_z (f(x, y, z) = 0)$ всюду определена, то функция g принадлежит также классу $\boldsymbol{R}_s$ (см. формулы, указанные в задаче 26). Из этого результата и результата, указанного в предыдущей задаче, вытекает следующая *теорема Поста о кванторном представлении*: если всюду определенные функции $f_1, \ldots, f_m$ принадлежат классу $\boldsymbol{R}_s$, то каждая всюду определенная функция, рекурсивная относительно $f_1, \ldots, f_s$, принадлежит тому же классу $\boldsymbol{R}_s$ (С. Клини [43], стр. 261).

29. Характеристическая функция любого предиката класса $\boldsymbol{R}_{s+1}$ рекурсивна относительно характеристических функций подходящих предикатов, имеющих классы $\boldsymbol{P}_s$ и соответственно $\boldsymbol{Q}_s$.

30. Пусть $\mathfrak{A}$ — замкнутая формула исчисления предикатов 1-й ступени, истинная на некоторой модели с бесконечным числом элементов и содержащая, помимо логических знаков (включая знак равенства), еще лишь предикатные символы $A_1, \ldots, A_m$. Тогда на натуральном ряде N существуют такие предикаты $A_1^0, \ldots, A_m^0$ класса R_2, что формула $\mathfrak{A}$ истинна на модели $\langle N; A_1^0, \ldots, A_m^0 \rangle$ (С. Клини [43], стр. 349).

31. Существуют удовлетворяющие условиям предыдущей задачи формулы $\mathfrak{A}$, которые ложны на любой модели $\langle \boldsymbol{N}; A_1^0, \ldots, A_m^0 \rangle$, предикаты которой $A_1^0, \ldots, A_m^0$ принадлежат классу $\boldsymbol{P}_1 \cup \boldsymbol{Q}_1$ (А. Мостовский [62]).

ГЛАВА VI

ВАРИАНТЫ МАШИН И АЛГОРИТМОВ ТЬЮРИНГА — ПОСТА

В этой главе будет изучено несколько классов алгоритмов, отличных от класса алгоритмов, выполняемых машинами Тьюринга, но довольно близких к последнему. В § 14 рассматриваются нормальные алгоритмы, операторные алгоритмы и так называемые продукции Поста, а в § 15 — многоленточные машины Тьюринга, двухленточные машины с неизменяющимися состояниями ячеек (машины Минского) и продукции Поста специального вида, известные под именем ТАГ-систем. Указанные классы алгоритмов наиболее часто встречаются в теоретических приложениях, если не считать алгоритмов, осуществляемых конечными автоматами, которые в данной книге не рассматриваются и составляют предмет особой дисциплины — теории конечных автоматов.

§ 14. НОРМАЛЬНЫЕ И ОПЕРАТОРНЫЕ АЛГОРИТМЫ

Каждое мгновенное состояние машины Тьюринга естественно описывается либо машинным словом, либо натуральным числом — номером машинного слова. В силу этих описаний переходу машины Тьюринга из одного состояния в другое отвечает либо определенное преобразование машинного слова, либо соответствующая операция над числом. Анализ преобразований слов непосредственно приводит к понятию алгоритма, заданного программой подстановок, описанному выше в п. 12.1, и к понятию нормального алгоритма, введенному А. А. Марковым [55]. Анализ операций над номерами машинных слов непосред-

ственно приводит к понятию операторного алгоритма Ван Хао [11]. Изложению всех этих понятий мы предпошлем несколько замечаний о еще более общем понятии формальной системы или формального исчисления, играющем важную роль в логике и других дисциплинах.

14.1. Формальные системы. Продукции Поста. Формальная система задается своим алфавитом $\boldsymbol{C} = \{c_1, c_2, \ldots, c_p\}$ и конечной совокупностью «правил вывода» $P_1, P_2, \ldots, P_s$. При этом правилом вывода P (n-местным) в алфавите $\boldsymbol{C}$ называется просто произвольная *рекурсивная* совокупность n-ок $(\mathfrak{x}_1, \ldots, \mathfrak{x}_n)$ слов в алфавите $\boldsymbol{C}$. Совокупность P обычно задают каким-нибудь «простым» правилом, позволяющим узнать, обладает произвольно взятая n-ка слов заданным признаком P или нет.

Слово $\mathfrak{x}$ называется непосредственным следствием из некоторой совокупности слов $\mathfrak{A}$, символически $\mathfrak{A} \models \mathfrak{x}$, если возможно указать такое правило вывода P_i и такие слова $\mathfrak{a}_1, \ldots, \mathfrak{a}_m$ в совокупности $\mathfrak{A}$, что последовательность $\langle \mathfrak{a}_1, \ldots, \mathfrak{a}_m, \mathfrak{x} \rangle$ принадлежит P_i.

Слово $\mathfrak{x}$ называется следствием из совокупности слов $\mathfrak{A}$, символически $\mathfrak{A} \vdash \mathfrak{x}$, если можно указать такую конечную последовательность слов $\mathfrak{x}_1, \ldots, \mathfrak{x}_t$, что

$$\mathfrak{A} \models \mathfrak{x}_1, \quad \{\mathfrak{A}, \mathfrak{x}_1\} \models \mathfrak{x}_2, \ldots, \{\mathfrak{A}, \mathfrak{x}_1, \ldots, \mathfrak{x}_t\} \models x.$$

Дополнительно полагают, что каждое слово есть непосредственное следствие самого себя. Таким образом, если $\mathfrak{x} \in \mathfrak{A}$, то $\mathfrak{A} \models \mathfrak{x}$.

Вместо «$\mathfrak{x}$ есть следствие из совокупности $\mathfrak{A}$» говорят также, что $\mathfrak{x}$ *выводимо из* $\mathfrak{A}$ (в данной формальной системе).

Из определения формальной выводимости непосредственно вытекают следующие ее свойства:

а) *Если* $\mathfrak{A} \vdash \mathfrak{x}$ *и* $\mathfrak{A} \subseteq \mathfrak{B}$, *то* $\mathfrak{B} \vdash \mathfrak{x}$.

б) *Если* $\mathfrak{A} \vdash \mathfrak{x}$ *и* $\{\mathfrak{A}, \mathfrak{x}\} \vdash \mathfrak{y}$, *то* $\mathfrak{A} \vdash \mathfrak{y}$.

в) *Если слово $\mathfrak{x}$ выводимо из некоторой бесконечной совокупности $\mathfrak{A}$ слов, то $\mathfrak{x}$ выводимо из некоторой конечной подсовокупности совокупности* $\mathfrak{A}$.

Примерами формальных систем могут служить ассоциативные исчисления, рассмотренные в п. 13.1. Пусть $\mathfrak{C}$ — ассоциативное исчисление с алфавитом $\boldsymbol{C}$ и определяющими соотношениями $\mathfrak{a}_i \equiv \mathfrak{b}_i$ $(i = 1, 2, \ldots, l)$. Каждому соотношению $\mathfrak{a}_i \equiv \mathfrak{b}_i$ ставим в соответствие два бинарных пра-

вила вывода: P_i и P'_i, полагая $P_i(\mathfrak{x}, \mathfrak{y})$ истинным, если слово $\mathfrak{y}$ может быть получено из $\mathfrak{x}$ подстановкой в $\mathfrak{x}$ слова $\mathfrak{b}_i$ вместо какого-нибудь вхождения $\mathfrak{a}_i$, и определяя аналогично $P'_i(\mathfrak{x}, \mathfrak{y})$. Ясно, что в формальной системе $\mathfrak{F}$ с алфавитом C и правилами вывода P_i, P'_i $(i = 1, 2, \ldots, l)$ $\mathfrak{x} \vdash \mathfrak{y}$ тогда и только тогда, когда $\mathfrak{x} \equiv \mathfrak{y}$ в заданном ассоциативном исчислении.

Пусть $\mathfrak{A}$ — некоторая совокупность слов формальной системы $\mathfrak{F}$. Формальным доказательством на основе $\mathfrak{A}$ называется каждая конечная цепочка $\langle \mathfrak{x}_0, \mathfrak{x}_1, \ldots, \mathfrak{x}_t \rangle$ слов из $\mathfrak{F}$, обладающая следующим свойством:

для каждого $\mathfrak{x}_i$, $0 \leqslant i \leqslant t$, или $\mathfrak{x}_i \in \mathfrak{A}$ или $\{\mathfrak{x}_0, \mathfrak{x}_1, \ldots$
$\ldots, \mathfrak{x}_{i-1}\} \models \mathfrak{x}_i$.

Последнее слово $\mathfrak{x}_t$ формального доказательства называется его *утверждением*, а все формальное доказательство $\langle \mathfrak{x}_0, \mathfrak{x}_1, \ldots, \mathfrak{x}_t \rangle$ называется формальным выводом $\mathfrak{x}_t$ из $\mathfrak{x}_0$ на основе $\mathfrak{A}$.

Теорема 1. *Пусть $\mathfrak{F}$ — какая-нибудь формальная система и $\mathfrak{A}$ — некоторое рекурсивно перечислимое множество слов в $\mathfrak{F}$. Тогда совокупность всех формальных доказательств в $\mathfrak{F}$ на основе $\mathfrak{A}$ также рекурсивно перечислима.*

Номером произвольной n-ки слов $\langle \mathfrak{x}_1, \ldots, \mathfrak{x}_n \rangle$ назовем канторовский номер числовой последовательности $\langle c(\mathfrak{x}_1), \ldots, c(\mathfrak{x}_n), n-1 \rangle$, где $c(\mathfrak{x}_i)$ — алфавитный номер слова $\mathfrak{x}_i$. Пусть $\alpha(m)$ — последовательность слов, имеющая номер m. По условию, существует примитивно рекурсивная функция $\varphi(x)$ такая, что $\varphi(0)$, $\varphi(1)$, ... суть алфавитные номера всех слов из $\mathfrak{A}$. Для каждой конечной последовательности слов $\langle \mathfrak{x}_0, \mathfrak{x}_1, \ldots, \mathfrak{x}_t \rangle$ мы можем решить вопрос о том, является ли она доказательством на основе совокупности $\mathfrak{A}_u$ слов с номерами $\varphi(0), \varphi(1), \ldots, \varphi(u)$. Теперь строим последовательность доказательств $\alpha(u_{00})$, $\alpha(u_{10})$, ... посредством следующего процесса.

а) Полагаем u_{00} равным номеру доказательства $\langle \mathfrak{a}, \mathfrak{a} \rangle$, где $\mathfrak{a}$ — слово с номером $\varphi(0)$.

б) Пусть доказательства

$$\alpha(u_{00}),\ \alpha(u_{10}), \ldots, \alpha(u_{1p_1}), \ldots, \alpha(u_{t0}), \ldots, \alpha(u_{tp_t}) \quad (1)$$

построены. Из последовательности цепочек слов $\alpha(0)$, $\alpha(1), \ldots, \alpha(t+1)$ выбираем те, которые являются

доказательствами на основе совокупности слов с номерами $\varphi(0)$, $\varphi(1)$, ..., $\varphi(t+1)$. Пусть это будут доказательства $\alpha(u_{t+1\,0})$, ..., $\alpha(u_{t+1\,p_{t+1}})$. Их мы и присоединяем к доказательствам (1).

Ясно, что так построенная последовательность доказательств содержит все доказательства на основе $\mathfrak{A}$. Члены этой последовательности строятся на основе ясного алгоритма и потому совокупность всех ее членов рекурсивно перечислима.

Следствие. *В любой формальной системе $\mathfrak{F}$ совокупность слов, выводимых из произвольного рекурсивно перечислимого множества слов $\mathfrak{A}$, является рекурсивно перечислимой.*

Действительно, согласно теореме 1 совокупность всех доказательств на основе $\mathfrak{A}$ рекурсивно перечислима. Но тогда рекурсивно перечислимыми являются совокупность всех доказательств, начальные слова которых находятся в $\mathfrak{A}$, и нужная нам совокупность всех последних слов доказательств, начинающихся словом из $\mathfrak{A}$.

Выше указывалось, что ассоциативные исчисления можно рассматривать как формальные системы определенного специального вида. Сейчас мы опишем формальные системы еще двух специальных видов: системы подстановок, называемые иногда полусистемами Туэ или системами полу-Туэ, и системы продукций.

Система подстановок (= подстановочная система) задается некоторым алфавитом $C = \{c_1, c_2, \ldots, c_p\}$ и базисными подстановками

$$\mathfrak{a}_i \to \mathfrak{b}_i \qquad (i = 1, \ldots, l), \tag{2}$$

где $\mathfrak{a}_i$, $\mathfrak{b}_i$ — некоторые (возможно и пустые) слова в алфавите C. Каждую подстановку $\mathfrak{a}_i \to \mathfrak{b}_i$ из (2) мы будем понимать как правила вывода P_i, считая, что $P_i(\mathfrak{x}, \mathfrak{y})$ истинно, если слово $\mathfrak{y}$ получается из слова $\mathfrak{x}$ посредством подстановки в $\mathfrak{x}$ слова $\mathfrak{b}_i$ вместо какого-нибудь вхождения слова $\mathfrak{a}_i$.

Отсюда непосредственно видно, что ассоциативное исчисление с алфавитом C и определяющими соотношениями

$$\mathfrak{a}_i \equiv \mathfrak{b}_i \qquad (i = 1, \ldots, l)$$

можно рассматривать как систему подстановок с базисными

подстановками

$$\mathfrak{a}_i \longrightarrow \mathfrak{b}_i, \quad \mathfrak{b}_i \longrightarrow \mathfrak{a}_i.$$

Таким образом, система подстановок (2) содержит как бы лишь «половину» подстановок, содержащихся в ассоциативном исчислении с соответствующими определяющими соотношениями.

Ассоциативные исчисления иногда называются системами Туэ в честь норвежского математика Акселя Туэ, в работах которого, по-видимому, впервые была отчетливо сформулирована проблема равенства слов в ассоциативных исчислениях. Это и дало повод системы подстановок называть *полусистемами Туэ* или даже «системами полу-Туэ».

Пусть $\mathfrak{S}$ — система подстановок с алфавитом C и базисными подстановками (2). Слово $\mathfrak{a}$ в алфавите C называется *заключительным в системе* $\mathfrak{S}$, если ни одно из левых слов $\mathfrak{a}_1, \ldots, \mathfrak{a}_l$ подстановок (2) не входит в $\mathfrak{a}$. Говорят, что система (2) *перерабатывает* слово $\mathfrak{x}$ в слово $\mathfrak{y}$, если $\mathfrak{x} \vdash \mathfrak{y}$ и слово $\mathfrak{y}$ заключительное. Может случиться, что для каждого слова $\mathfrak{x}$ из некоторого рекурсивного множества $\mathfrak{M}$ существует не более одного заключительного слова $\mathfrak{y}$, удовлетворяющего соотношению $\mathfrak{x} \vdash \mathfrak{y}$. Тогда, ставя слову $\mathfrak{x}$ в соответствие слово $\mathfrak{y}$, получим частичную функцию на $\mathfrak{M}$, которая называется функцией, вычисляемой системой $\mathfrak{S}$. Поскольку множество пар слов $\langle \mathfrak{x}, \mathfrak{y} \rangle$, удовлетворяющих соотношению $\mathfrak{x} \vdash \mathfrak{y}$, рекурсивно перечислимо, а множество всех заключительных слов рекурсивно, то упомянутая функция заведомо частично рекурсивна. Программа подстановок, отвечающая машине Тьюринга, показывает, что при надлежащем кодировании чисел при помощи систем подстановок можно вычислить любую частично рекурсивную функцию.

Система продукций Поста задается своим алфавитом $C = \{c_1, \ldots, c_p\}$ и системой базисных продукций

$$\mathfrak{a}_i W \longrightarrow W \mathfrak{b}_i \qquad (i = 1, \ldots, l), \tag{3}$$

где $\mathfrak{a}_i$, $\mathfrak{b}_i$ — какие-то слова в алфавите C. Пусть некоторое слово $\mathfrak{x}$ начинается словом $\mathfrak{a}_i$. Произвести над $\mathfrak{x}$ продукцию $\mathfrak{a}_i W \to W \mathfrak{b}_i$ — это значит вычеркнуть из $\mathfrak{x}$ начальный отрезок $\mathfrak{a}_i$ и затем к оставшемуся слову приписать справа слово $\mathfrak{b}_i$. Например, произведя над словом aba продукцию $abW \to Wc$, получим слово ac.

Каждая система продукций (3) обычно понимается как формальная система с правилами вывода P_i ($i = 1, \ldots, l$), где $P_i(\mathfrak{x}, \mathfrak{y})$ считается истинным, если слово $\mathfrak{y}$ получается из $\mathfrak{x}$ при помощи продукции $\mathfrak{a}_i W \to W\mathfrak{b}_i$.

Следующая простая теорема показывает, что любую систему подстановок можно «вложить» в систему продукций.

Теорема 2. *Пусть задана система подстановок $\mathfrak{S}$ с базисными подстановками* (2) *и алфавитом* $C = \{c_1, \ldots, c_p\}$. *Рассмотрим новый алфавит* $D = \{c_1, \ldots, c_p, c'_1, \ldots, c'_p\}$. *Пусть $\mathfrak{a}'$ обозначает слово, получающееся из слова $\mathfrak{a}$ в C заменой всех его букв соответствующими штрихованными буквами* (*например*, $(c_1c_2)' = c'_1c'_2$). *Обозначим через $\mathfrak{P}$ систему продукций с алфавитом D и базисными продукциями* 169

$$\begin{aligned} &\mathfrak{a}_i W \to W\mathfrak{b}'_i && (i = 1, \ldots, l), \\ &c_j W \to W c'_j, && \\ &c'_j W \to W c_j && (j = 1, \ldots, p). \end{aligned} \qquad (4)$$

Для любых двух слов $\mathfrak{x}$, $\mathfrak{y}$ в алфавите C отношение $\mathfrak{x} \vdash \mathfrak{y}$ истинно в системе подстановок $\mathfrak{S}$ тогда и только тогда, когда оно истинно в системе продукций $\mathfrak{P}$.

В самом деле, непосредственно ясно, что если слово $\mathfrak{y}$ в C получается из слова $\mathfrak{x}$ подстановкой $\mathfrak{a}_i \to \mathfrak{b}_i$, то $\mathfrak{y}$ можно получить из $\mathfrak{x}$ совокупностью продукций вида (4). Поэтому из $\mathfrak{x} \vdash \mathfrak{y}$ в $\mathfrak{S}$ следует $\mathfrak{x} \vdash \mathfrak{y}$ в $\mathfrak{P}$.

Обратно, пусть для некоторых слов $\mathfrak{x}$, $\mathfrak{y}$ в алфавите C имеем $\mathfrak{x} \vdash \mathfrak{y}$ в $\mathfrak{P}$. Тогда найдется конечная цепочка слов $\mathfrak{x}_1, \mathfrak{x}_2, \ldots, \mathfrak{x}_t$ в алфавите D такая, что в цепочке

$$\mathfrak{x} = \mathfrak{x}_0, \mathfrak{x}_1, \ldots, \mathfrak{x}_t, \mathfrak{x}_{t+1} = \mathfrak{y}$$

каждое слово $\mathfrak{x}_{i+1}$ получается из предыдущего слова $\mathfrak{x}_i$ посредством какой-то из продукций (4). Индукцией по i легко убеждаемся сначала, что каждое слово $\mathfrak{x}_i$ имеет вид $\mathfrak{x}_i = \mathfrak{y}_i\mathfrak{z}'_i$ или $\mathfrak{x}_i = \mathfrak{z}'_i\mathfrak{y}_i$, где $\mathfrak{y}_i$, $\mathfrak{z}_i$ — слова в алфавите C. Полагаем $(\mathfrak{y}\mathfrak{z}')^* = (\mathfrak{y}'\mathfrak{z})^* = \mathfrak{z}\mathfrak{y}$. Просматривая продукции (4), сразу же видим, что для каждого i либо $(\mathfrak{x}_i)^* = (\mathfrak{x}_{i+1})^*$, либо $(\mathfrak{x}_{i+1})^*$ получается из $(\mathfrak{x}_i)^*$ одной из подстановок (2), что и требовалось.

Следствие. *Существует такая система продукций $\mathfrak{P}$, в которой совокупность всех слов, выводимых из подходящего фиксированного слова $\mathfrak{a}$, является нерекурсивной.*

В самом деле, пусть $A(\mathfrak{T})$ — ассоциативное исчисление с алфавитом $\{0, 1, q_0, q_1, \ldots, q_n, v\} = \boldsymbol{C}$, в котором неразрешима проблема равенства слову $q_0 v$. Такое исчисление было построено в п. 13.1. Обозначим через $\mathfrak{P}$ систему продукций с алфавитом $\boldsymbol{D} = \{0, 1, q_0, \ldots, q_n, v, 0', 1', q_0', \ldots, q_n', v'\}$, получаемую из ассоциативного исчисления процессом, описанным в теореме 2. Тогда совокупность тех слов в алфавите $\boldsymbol{C}$, которые выводимы продукциями из слова $q_0 v$, будет совпадать с совокупностью всех слов, эквивалентных слову $q_0 v$ в исчислении $A(\mathfrak{T})$. Так как последняя совокупность нерекурсивна, то нерекурсивна и совокупность всех слов в алфавите $\boldsymbol{D}$, выводимых продукциями из $q_0 v$.

14.2. Нормальные алгоритмы. Фиксируем какой-нибудь алфавит $\boldsymbol{A}$ и пусть символы $\rightarrow$ и $\cdot$ не входят в $\boldsymbol{A}$. Следуя А. А. Маркову, формулы вида

$$\mathfrak{a} \rightarrow \mathfrak{b}, \qquad \mathfrak{a} \rightarrow \cdot\mathfrak{b}, \tag{1}$$

где $\mathfrak{a}$, $\mathfrak{b}$ — какие-нибудь слова (возможно и пустые) в алфавите $\boldsymbol{A}$, будем называть соответственно *простыми* и *заключительными* формулами подстановок. Слово $\mathfrak{a}$ будет называться левым, а слово $\mathfrak{b}$ — правым словом формулы.

Произвольная конечная последовательность формул подстановок называется *схемой. Нормальным алгоритмом с данной схемой* $\mathfrak{A}$ в алфавите $\boldsymbol{A}$ называется следующее предписание для переработки слов в алфавите $\boldsymbol{A}$.

Пусть задано произвольное слово $\mathfrak{x}$ в алфавите $\boldsymbol{A}$. Полагаем по определению $\mathfrak{x}_0 = \mathfrak{x}$ и говорим, что слово $\mathfrak{x}_0$ получается из слова $\mathfrak{x}$ после нуля шагов переработки. Пусть для некоторого натурального n нам уже известно слово $\mathfrak{x}_n$, полученное из $\mathfrak{x}$ после n шагов переработки. Тогда $(n+1)$-й шаг переработки будет состоять в следующих действиях. В схеме $\mathfrak{A}$ ищем первую формулу подстановок (1), левая часть которой $\mathfrak{a}$ есть подслово слова $\mathfrak{x}_n$, и затем в $\mathfrak{x}_n$ вместо первого вхождения $\mathfrak{a}$ подставляем правое слово $\mathfrak{b}$ этой формулы. Возникшее в результате подстановки слово обозначается через $\mathfrak{x}_{n+1}$. Если использованная для получения $\mathfrak{x}_{n+1}$ формула подстановок была простой, то говорим, что $\mathfrak{x}_{n+1}$ есть слово, получающееся из $\mathfrak{x}_n$ после $n+1$ шагов переработки. Если же указанная формула подстановки была заключительной, то слово $\mathfrak{x}_{n+}$

называется *результатом* переработки слова $\mathfrak{x}$ посредством алгоритма $\mathfrak{A}$ и обозначается символом $\mathfrak{A}(\mathfrak{x})$.

Может случиться, что в схеме $\mathfrak{A}$ не окажется ни одной формулы подстановок, левое слово которой входит в качестве подслова в слово $\mathfrak{x}_n$. В этом случае мы полагаем $\mathfrak{x}_{n+1} = \mathfrak{x}_n$ и также говорим, что $\mathfrak{x}_{n+1}$ есть результат переработки слова $\mathfrak{x}$ посредством алгоритма $\mathfrak{A}$. В обоих последних случаях говорят, что процесс переработки слова *обрывается* после $(n+1)$-го шага. Если процесс переработки слова $\mathfrak{x}$ ни на каком шаге не обрывается, то говорят, что результат переработки слова $\mathfrak{x}$ посредством алгоритма *не определен*. В этом последнем случае символу $\mathfrak{A}(\mathfrak{x})$ приписывается *неопределенное* значение.

В формулах подстановок (1) слова $\mathfrak{a}$, $\mathfrak{b}$ могут быть пустыми. В связи с этим возникает вопрос, что значит в данное слово $\mathfrak{x}$ подставить $\mathfrak{b}$ вместо первого вхождения в $\mathfrak{x}$ пустого слова $\mathfrak{a} = \Lambda$? По определению полагают, что результатом указанной подстановки является слово $\mathfrak{b}\mathfrak{x}$. Например, нормальный алгоритм $\mathfrak{A}$ со схемой, состоящей лишь из одной формулы $\Lambda \to \cdot\mathfrak{a}$, каждое слово $\mathfrak{x}$ перерабатывает в слово $\mathfrak{a}\mathfrak{x}$.

Выше определено понятие нормального алгоритма в алфавите $\boldsymbol{A}$. Если алгоритм $\mathfrak{A}$ задан в некотором расширении алфавита $\boldsymbol{A}$, то говорят, что $\mathfrak{A}$ есть нормальный алгоритм *над* $\boldsymbol{A}$. Одноместная частичная словарная функция $F(\mathfrak{x})$, заданная в алфавите $\boldsymbol{A}$, называется *нормально вычислимой*, если существует нормальный алгоритм $\mathfrak{A}$ над алфавитом $\boldsymbol{A}$ такой, что для каждого слова $\mathfrak{x}$ в алфавите $\boldsymbol{A}$ выполнено равенство $F(\mathfrak{x}) = \mathfrak{A}(\mathfrak{x})$.

В частности, алгоритм со схемой $\Lambda \to \cdot\Lambda$ вычисляет функцию $F(\mathfrak{x}) = \mathfrak{x}$, а алгоритм со схемой $\Lambda \to \Lambda$ вычисляет нигде не определенную функцию.

В качестве более содержательного примера рассмотрим словарную функцию в алфавите 0, 1, заданную формулой $F(\mathfrak{x}) = \mathfrak{x}\mathfrak{a}$. Пусть $\mathfrak{A}$ — нормальный алгоритм в более широком алфавите $\{0, 1, 2\}$, имеющий схему

$$20 \to 02, \quad 21 \to 12, \quad 2 \to \cdot\mathfrak{a}, \quad \Lambda \to 2.$$

Возьмем какое-нибудь слово в алфавите 0, 1, например, слово $0110 = \mathfrak{x}$. После первого шага мы получим слово 20110. Каждый следующий шаг переработки будет сдвигать

символ 2 на одно место вправо и после 5-го шага мы будем иметь слово 01102, которое после использования заключительной формулы даст слово 0110 $\mathfrak{a} = \mathfrak{x}\mathfrak{a}$.

Таким образом, алгоритм *над* алфавитом {0, 1} с указанной схемой вычисляет функцию $\mathfrak{x}\mathfrak{a}$, заданную в алфавите {0, 1}.

Мы не будем рассматривать более сложные примеры, так как легко устанавливается следующая общая

Теорема 1 (Детловс [33]). *Класс нормально вычислимых частичных функций, заданных в произвольном алфавите* $\boldsymbol{A}$, *совпадает с классом всех одноместных частично рекурсивных словарных функций в алфавите* $\boldsymbol{A}$.

Для доказательства достаточно сравнить определение нормального алгоритма с определением алгоритма Тьюринга при помощи программы подстановок. В самом деле, пусть $F(\mathfrak{x})$ — произвольная частично рекурсивная словарная функция, заданная в алфавите $\boldsymbol{A}$. Согласно п. 12.3 существует машина Тьюринга $\mathfrak{T}$, имеющая внешний алфавит a_0, $\boldsymbol{A}$ (a_0 — новый, не входящий в $\boldsymbol{A}$ символ) и вычисляющая функцию $F(\mathfrak{x})$.

Пусть q_0, q_1, . . ., q_n — внутренние состояния $\mathfrak{T}$ и $\mathfrak{P}$ — программа подстановок для $\mathfrak{T}$. Заменяя входящие в $\mathfrak{P}$ формулы вида $q_i a_j \to q_0 a_k$ формулами $q_i a_j \to \cdot\, q_0 a_k$, получим схему некоторого нормального алгоритма $\mathfrak{A}$ в алфавите $\{a_0, A, q_0, \ldots, q_n\}$. Сравнивая описания работы машины $\mathfrak{T}$ и операций, предписываемых алгоритмом $\mathfrak{A}$, непосредственно видим, что для каждого слова $\mathfrak{x}$ в алфавите $\boldsymbol{A}$ $\mathfrak{P}(\mathfrak{x}) = \mathfrak{A}(\mathfrak{x})$. Таки образом, каждая частично рекурсивная словарная функция нормально вычислима.

Обратно, пусть задан нормальный алгоритм со схемой $\mathfrak{A}$ в алфавите $\boldsymbol{A}$. Пусть символ q_0 не входит в $\boldsymbol{A}$. Заменяя точки в формулах $\mathfrak{A}$ символом q_0, присоединяя в начале формулу $q_0 \to q_0$ и в конце формулу $\Lambda \to q_0$, получим программу подстановок (в смысле п. 12.2). В п. 12.2 доказано, что частичная словарная функция $F_{\mathfrak{A}}^{l}(\mathfrak{x})$, заданная в алфавите A и вычисляемая с помощью указанной программы подстановок, является частично рекурсивной. Так как ясно, что функции $F_{\mathfrak{A}}(\mathfrak{x})$ и $\mathfrak{A}(\mathfrak{x})$ совпадают, то мы приходим к выводу, что каждая нормально вычислимая функция частично рекурсивна.

14.3. Операторные алгоритмы. Операторный алгоритм задается последовательностью приказов, каждый из которых имеет определенный номер и содержит указания: какую операцию следует выполнить над заданным объектом и приказ с каким номером следует далее выполнять над результатом данной операции. В качестве объектов мы будем брать натуральные числа, а приказы будем рассматривать вида

$$i: \boxed{w \mid \alpha \mid \beta}\,, \qquad (1)$$

где i — номер приказа, w — символ фиксированной одноместной частичной функции, α, β — номера некоторых приказов. Выполнить приказ (1) над числом x — значит найти число $w(x)$ и далее перейти к выполнению над $w(x)$ приказа с номером α. Если же $w(x)$ не определено, то перейти к выполнению над числом x приказа с номером β. Например, выполнив над числом 24 приказ

$$i: \boxed{:6 \mid 7 \mid 13}\,,$$

получим число 4 и указание выполнять далее над 4 приказ с номером 7. Выполнив же упомянутый приказ над числом 23, получим снова число 23 и указание выполнять над 23 приказ с номером 13.

Помимо приказов вида (1), будет нужен еще приказ

$$i: \boxed{\text{стоп}}\,, \qquad (2)$$

который означает, что вычисления следует остановить и что число, над которым следует выполнить приказ (2), есть *результат* вычислений.

Результат выполнения приказа (1) над числом x мы будем изображать парой (z, γ), где z — полученное число, γ — номер приказа, который должен выполняться далее над z.

Программой операторного алгоритма называется последовательность приказов вида

$$0: \boxed{\text{стоп}}\,,\quad 1: \boxed{w_1 \mid \alpha_1 \mid \beta_1}\,, \ldots,\quad s: \boxed{w_s \mid \alpha_s \mid \beta_s}\,, \qquad (3)$$

где $\omega_1(x), \ldots, \omega_s(x)$ — заданные частичные функции, $\alpha_1, \beta_1, \ldots, \alpha_s, \beta_s$ — какие-то натуральные числа из последовательности $0, 1, \ldots, s$.

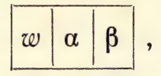
Пусть задано произвольное число x. Переработать x согласно указанной программе — это значит выполнить над x последовательность следующих действий:

1-й шаг: находим $\omega_1(x)$. Если $\omega_1(x)$ определено, то результатом первого шага будет пара чисел $(\omega_1(x), \alpha_1)$. Если $\omega_1(x)$ не определено, то результатом первого шага будет пара (x, β_1).

2-й шаг: пусть пара (x_1, γ) есть результат предыдущего шага. Выполняем над x_1 приказ с номером γ из программы (3), то есть находим $\omega_\gamma(x_1)$. Если $\omega_\gamma(x_1)$ определено, то результатом 2-го шага будет пара $(\omega_\gamma(x_1), \alpha_\gamma)$. Если же $\omega_\gamma(x_1)$ не определено, то результатом 2-го шага будет пара (x_1, β_γ) и т. д.

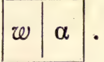
Если на n-м шаге получится пара вида $(a, 0)$, то по определению на этом процесс обрывается и число a называется *результатом переработки* первоначального числа x согласно программе (3). Если же пара вида $(a, 0)$ ни на каком шаге не возникает, то результатом переработки x будет «неопределенное значение».

В дальнейшем мы всегда будем приписывать номер 0 приказу «стоп» и номер 1 начальному приказу. Остальные же приказы иногда будем нумеровать не только натуральными числами, но и парами чисел, и символами какого-нибудь алфавита. Важно лишь одно: если программа содержит приказ



,

то она должна содержать и приказы с номерами α, β. Если в упомянутом приказе функция ω всюду определенная, то символ β не оказывает влияния на течение вычислений и потому обычно не указывается. Тогда пишут и так:



.

Говорят, что операторный алгоритм A с программой (3) вычисляет частичную функцию $f(x)$, если алгоритм A перерабатывает каждое натуральное число x в $f(x)$. В частности, если $f(x)$ не определено, то процесс переработки x согласно программе (3) должен быть бесконечным.

Природа функций, вычислимых посредством операторных алгоритмов, зависит, конечно, от того, какие функции w_i входят в записи приказов. Простейшим фактом в этом направлении является

Теорема 1. *Для того чтобы частичная функция $f(x)$ была вычислима с помощью операторного алгоритма, программа которого* (3) *содержит лишь частично рекурсивные функции $w_i(x)$ с рекурсивной областью определенности, необходимо и достаточно, чтобы $f(x)$ была частично рекурсивной.*

Необходимость условий очевидна, и мы ограничимся лишь доказательством достаточности их. Ясно, что алгоритм с программой

$$0: \boxed{\text{стоп}}\ ; \qquad 1: \boxed{+2 \mid 1 \mid \quad}$$

вычисляет нигде не определенную функцию $f(x)$.

Пусть частично рекурсивная функция $f(x)$ имеет непустой график. Тогда этот график можно представить в виде совокупности пар $\langle \alpha(t), \beta(t) \rangle$ $(t = 0, 1, \ldots)$, где α, β — подходящие примитивно рекурсивные функции. Обозначим через $v(x)$ выражение 2^x и введем частичную функцию $w(x)$, полагая

$$w(x) = \begin{cases} 3x, & \text{если } \mathrm{ex}_0\, x \neq \alpha(\mathrm{ex}_1\, x), \\ \text{неопр.}, & \text{если } \mathrm{ex}_0\, x = \alpha(\mathrm{ex}_1\, x). \end{cases}$$

Функция $w(x)$ частично рекурсивна с рекурсивной областью определенности. Легко убедиться, что операторный алгоритм, имеющий программу:

$$0: \boxed{\text{стоп}}\ ; \quad 1: \boxed{v \mid 2}\ ; \quad 2: \boxed{w \mid 2 \mid 3}\ ;$$

$$3: \boxed{\mathrm{ex}_1 \mid 4}\ ; \quad 4: \boxed{\beta \mid 0}\ ,$$

вычисляет как раз функцию $f(x)$. В самом деле, выполнив

над произвольным числом x приказ 1, получим пару $(2^x, 2)$. Если $x \neq \alpha(0)$, то, выполнив над 2^x приказ 2, получим пару $(2^x \cdot 3^1, 2)$. Если $x \neq \alpha(1)$, то, выполнив над $2^x \cdot 3^1$ приказ 2, получим пару $(2^x \cdot 3^2, 2)$ и т. д. Процесс продолжается до тех пор, пока, наконец, получится пара $(2^x \cdot 3^n, 2)$, для которой $x = \alpha(n)$. Так как $\varpi(2^x \cdot 3^n)$ не определено, то над числом $2^x \cdot 3^n$ теперь надо выполнить приказ 3, в результате которого получим пару $(n, 4)$. Выполнив над n приказ 4, получим пару $(\beta(n), 0)$. Так как $x = \alpha(n)$, то $\beta(n) = f(x)$ и, следовательно, алгоритм перерабатывает x в $f(x)$, что и требовалось.

Построенная выше программа для вычисления функции $f(x)$ оказалась довольно тривиальной вследствие того, что запас функций, допустимых в приказах, был очень большой. Естественно, чем у́же запас, тем труднее строить нужные программы. Поэтому несомненный интерес представляет следующая теорема, машинную интерпретацию которой мы рассмотрим далее в п. 15.2.

Теорема 2 (Минский [60]). *Для каждой частично рекурсивной функции $f(x)$ существует операторный алгоритм, программа которого состоит из приказов вида*

стоп

,

$\times c$	α	

,

$:d$	α	β

(4)

$$(c = 2, 3, 5;\ d = 2 \cdot 3 \cdot 5),$$

для любого x перерабатывающий 2^x в $2^{f(x)}$.

Иными словами, любая частично рекурсивная функция $f(x)$ вычислима при помощи подходящего алгоритма, программа которого состоит из приказов вида (4), при условии, что значения аргумента и функции кодируются числами 2^x, $2^{f(x)}$. Из доказательства теоремы будет видно, что в ее формулировке числа 3, 5 можно заменить любыми другими взаимно простыми нечетными числами.

Прежде всего мы покажем, как для любой машины Тьюринга $\mathfrak{T}$ с внешним алфавитом 0, 1 можно построить программу операторного алгоритма, выполняющего, грубо говоря, ту же работу, что и машина $\mathfrak{T}$. Мы можем предполагать, что в процессе работы машина $\mathfrak{T}$ не надстраивает ленты слева. Пусть $q_0, q_1, \ldots, q_n$ — внутренние состояния

машины $\mathfrak{T}$. Мгновенное состояние всей машины $\mathfrak{T}$ описывается машинным словом

$$\mathfrak{x} = b_r b_{r-1} \ldots b_0 q_i a_0 a_1 \ldots a_s \qquad (b_\lambda,\ a_\mu = 0,\ 1), \tag{5}$$

которое мы условимся кодировать парой чисел $(2^a \cdot 3^b,\ i)$, где

$$\begin{aligned} a &= a_0 + a_1 \cdot 2 + \ldots + a_s \cdot 2^s, \\ b &= b_0 + b_1 \cdot 2 + \ldots + b_r \cdot 2^r. \end{aligned} \tag{6}$$

В зависимости от i и a_0 машина $\mathfrak{T}$ перейдет из конфигурации $\mathfrak{x}$ в конфигурацию $\mathfrak{x}^1$. Пусть конфигурация $\mathfrak{x}^1$ кодируется парой $(2^c \cdot 3^d,\ j)$. Число $2^c \cdot 3^d$ зависит от чисел $2^a \cdot 3^b$, i, a_0, а число j зависит лишь от i и a_0. Поэтому полагаем $j = h(i,\ a_0)$.

Но число a_0 само есть функция от числа $z = 2^a \cdot 3^b$:

$$a_0 = \begin{cases} 0, & \text{если } \mathrm{ex}_0 z \text{ четное}, \\ 1, & \text{если } \mathrm{ex}_0 z \text{ нечетное}. \end{cases}$$

Следовательно, c и d будут зависеть от $z = 2^a \cdot 3^b$ и i, и мы можем ввести для каждого $i = 1, \ldots, n$ две частичные функции w_{i0}, w_{i1}, полагая по определению

$$\begin{aligned} w_{i0}(z) &= \begin{cases} 2^c \cdot 3^d, & \text{если } \mathrm{ex}_0 z \text{ четное}, \\ \text{неопр.}, & \text{если } \mathrm{ex}_0 z \text{ нечетное}, \end{cases} \\ w_{i1}(z) &= \begin{cases} 2^c \cdot 3^d, & \text{если } \mathrm{ex}_0 z \text{ нечетное}, \\ \text{неопр.}, & \text{если } \mathrm{ex}_0 z \text{ четное}. \end{cases} \end{aligned} \tag{7}$$

Рассмотрим операторный алгоритм $\boldsymbol{A}$ с программой

$$\begin{array}{ll} 0: \boxed{\text{стоп}} & \\ 1: \boxed{w_{10} \mid h(1,\ 0) \mid 1^*} & 1^*: \boxed{w_{11} \mid h(1,\ 1) \mid \quad}, \\ \ldots\ldots\ldots\ldots\ldots\ldots & \ldots\ldots\ldots\ldots\ldots\ldots \\ n: \boxed{w_{n0} \mid h(n,\ 0) \mid n^*} & n^*: \boxed{w_{n1} \mid h(n,\ 1) \mid \quad}. \end{array} \tag{8}$$

Пара приказов

$$i: \boxed{w_{i0} \mid h(i,\ 0) \mid i^*}\ ; \qquad i^* \boxed{w_{i1} \mid h(i,\ 1) \mid \quad}$$

переводит указанную выше пару чисел $(2^a \cdot 3^b,\ i)$, отвечающих конфигурации $\mathfrak{x}$, в пару чисел $(2^c \cdot 3^d,\ j)$, отвечающих конфигурации $\mathfrak{x}^1$. В частности, если машина $\mathfrak{T}$ перерабатывает какую-нибудь начальную конфигурацию

$$b_r b_{r-1} \ldots b_0 q_1 a_0 a_1 \ldots a_s \tag{9}$$

в заключительную конфигурацию

$$d_u d_{u-1} \ldots d_0 q_0 c_0 c_1 \ldots c_v, \tag{10}$$

то алгоритм A с программой (8) переработает число $2^a \cdot 3^b$ в число $2^c \cdot 3^d$, где a, b выражаются формулами (6), а c, d вычисляются по таким же формулам для (10).

Приказы программы (8) содержат более сложные функции, чем те, которые указываются теоремой 2. Наша задача состоит в том, чтобы каждый приказ из программы (8) заменить равносильной ему серией приказов вида (4).

Рассмотрим приказ i: $\boxed{w_{i0} \mid j \mid i^*}$. Нам задана пара чисел $(2^a \cdot 3^b,\ i)$, кодирующая некоторую конфигурацию (5) машины $\mathfrak{T}$. В соответствие с формулой (7) мы должны прежде всего узнать, четно a или нет. Для этого пишем серию приказов

i: $\boxed{:4 \mid i.1 \mid i.2}$; $i.1$: $\boxed{\times 5 \mid i \mid \ }$; $i.2$: $\boxed{:2 \mid i.3 \mid i.4}$.

В результате этих приказов пара $(2^a \cdot 3^b,\ i)$ преобразуется в пару $(3^b \cdot 5^{\frac{a-1}{2}},\ i.3)$, если a нечетное, и в пару $(3^b \cdot 5^{\frac{a}{2}},\ i.4)$, если a четное. В первом случае нам надо далее перейти к паре $(2^a \cdot 3^b,\ i^*)$. Очевидно, мы достигнем этого приказами

$i.3$: $\boxed{:5 \mid i.5 \mid i.6}$; $i.5$: $\boxed{\times 4 \mid i.3}$; $i.6$: $\boxed{\times 2 \mid i^*}$.

Во втором случае нам надо от пары $(3^b \cdot 5^{\frac{a}{2}},\ i.4)$ перейти к паре $(2^c \cdot 3^a,\ j)$, которая кодирует машинное слово $\mathfrak{x}^1$, непосредственно следующее за $\mathfrak{x}$. Чтобы найти $\mathfrak{x}^1$, надо обратиться к программе машины $\mathfrak{T}$. Так как обозреваемая ячейка a_0 находится в состоянии 0, то рассматриваем

ту из команд

$$q_i 0 \to q_j e, \quad q_i 0 \to q_j L, \quad q_i 0 \to q_j R,$$

которая содержится в программе $\mathfrak{T}$. Пусть, например, это команда $q_i 0 \to q_j R$. Тогда

$$\mathfrak{x}^1 = b_r b_{r-1} \ldots b_0 0 q_j a_1 \ldots a_s$$

и кодирующей парой является пара $(2^{\frac{a}{2}} \cdot 3^{2b}, j)$.

Чтобы перейти к этой паре от пары $(3^b \cdot 5^{\frac{a}{2}}, i.4)$, пишем приказы

$i.4$: | :5 | $i.7$ | $i.8$ | ; $i.7$: | $\times 2$ | $i.4$ | ,

в результате которых из пары $(3^b \cdot 5^{\frac{a}{2}}, i.4)$ получим пару $2^{\frac{a}{2}} \cdot 3^b$, $i.8$). Приказами

$i.8$: | :3 | $i.9$ | $i.10$ | ; $i.9$: | $\times 5$ | $i.8$ |

переводим пару $(2^{\frac{a}{2}} \cdot 3^b, i.8)$ в пару $(2^{\frac{a}{2}} \cdot 5^b, i.10)$. Затем приказами

i 10: | :5 | $i.11$ | j | ; $i.11$: | $\times 9$ | $i.10$ |

переводим пару $(2^{\frac{a}{2}} \cdot 5^b, i.10)$ в требуемую пару $(2^{\frac{a}{2}} \cdot 3^{2b}, j)$.

Итак, мы указали, как построить серию новых приказов i, $i.1$, ..., $i.11$, равносильную старому приказу i, в случае, когда машина $\mathfrak{T}$ выполняет команду $q_0 0 \to q_j R$. Ясно, что тем же способом соответствующие серии приказов можно построить и в тех случаях, когда машина $\mathfrak{T}$ вместо команды $q_i 0 \to q_j R$ выполняет команду $q_i 0 \to q_j L$ или команду $q_i 0 \to q_j e$.

Наконец, тем же способом строится и серия новых приказов, заменяющая старый приказ с номером i^*.

Среди новых приказов все еще встречаются приказы, имеющие не тот вид, который требуется теоремой 2.

Но от этого недостатка легко избавиться. Например, приказ γ: $\begin{array}{|c|c|}\hline \times 9 & \alpha \\ \hline \end{array}$ равносилен паре приказов

$$\gamma\colon \begin{array}{|c|c|}\hline \times 3 & \gamma^* \\ \hline \end{array}\ ; \quad \gamma^*\colon \begin{array}{|c|c|}\hline \times 3 & \alpha \\ \hline \end{array}\ ,$$

а приказ вида γ: $\begin{array}{|c|c|c|}\hline :2 & \alpha & \beta \\ \hline \end{array}$ равносилен приказам

$$\gamma\colon \begin{array}{|c|c|}\hline \times 3 & \gamma.0 \\ \hline \end{array}\ ; \quad \gamma.0\colon \begin{array}{|c|c|}\hline \times 5 & \gamma.1 \\ \hline \end{array}\ ; \quad \gamma.1\colon \begin{array}{|c|c|c|}\hline :30 & \alpha & \gamma.2 \\ \hline \end{array}\ ;$$

$$\gamma.2\colon \begin{array}{|c|c|}\hline \times 2 & \gamma.3 \\ \hline \end{array}\ ; \quad \gamma.3\colon \begin{array}{|c|c|}\hline :30 & \beta \\ \hline \end{array}\ ,$$

имеющим уже стандартную форму, требуемую теоремой 2. То же относится и к приказам

$$\begin{array}{|c|c|}\hline \times 4 & \alpha \\ \hline \end{array}\ , \quad \begin{array}{|c|c|c|}\hline :5 & \alpha & \beta \\ \hline \end{array}\ , \quad \begin{array}{|c|c|c|}\hline :3 & \alpha & \beta \\ \hline \end{array}\ .$$

После всех указанных замен у нас получится операторный алгоритм $\boldsymbol{B}$, программа которого состоит из приказов вида, требуемого теоремой 2. Из построения видно, что новый алгоритм $\boldsymbol{B}$ обладает тем же основным свойством, что и алгоритм $\boldsymbol{A}$, а именно, если машина $\mathfrak{T}$ из какой-нибудь начальной конфигурации (9) переходит через конечное число тактов работы в заключительную конфигурацию (10), то алгоритм $\boldsymbol{B}$ перерабатывает число $2^a \cdot 3^b$ в число $2^c \cdot 3^d$.

Пусть теперь нам задана какая-нибудь частично рекурсивная функция $f(x)$. В соответствии с п. 12.3 строим машину Тьюринга $\mathfrak{T}$, имеющую внешний алфавит 0, 1 и перерабатывающую машинное слово $q_1 01^x$ в слово $q_0 01^{f(x)}$. Обозначим через **B** операторный алгоритм, отвечающий машине $\mathfrak{T}$ и имеющий программу вида (4). Так как упомянутые машинные слова кодируются парами

$$(2^{2^{x+1}-2},\ 1), \quad (2^{2^{f(x)+1}-2},\ 0),$$

то алгоритм **B** перерабатывает число $2^{2^{x+1}-2}$ в число

$2^{2^{f(x)+1}-2}$. Мы сначала изменим алгоритм **B** так, чтобы он переводил 2^{2^x} в $2^{2^{f(x)}}$.

Легко видеть, что алгоритм **C**, имеющий программу

$$1\colon \boxed{:2 \mid 2 \mid 3}\,; \quad 2\colon \boxed{\times 3 \mid 1}\,; \quad 3\colon \boxed{:3 \mid 4 \mid 5}\,;$$

$$4\colon \boxed{\times 4 \mid 3}\,; \quad 5\colon \boxed{:4 \mid 0}$$

перерабатывает число 2^{2^x} в число $2^{2^{x+1}-2}$, а алгоритм **D**, имеющий программу

$$1\colon \boxed{\times 4 \mid 2}\,; \quad 2\colon \boxed{:4 \mid 3 \mid 4}\,; \quad 3\colon \boxed{\times 3 \mid 2}\,;$$

$$4\colon \boxed{:3 \mid 5 \mid 0}\,; \quad 5\colon \boxed{\times 2 \mid 4}\,,$$

перерабатывает $2^{2^{x+1}-2}$ в 2^{2^x}. Таким образом, алгоритм **CBD** перерабатывает 2^{2^x} в $2^{2^{f(x)}}$.

Нам остается построить программы алгоритмов, перерабатывающих 2^x в 2^{2^x} и соответственно 2^{2^x} в 2^x. Приказы

$$1\colon \boxed{:2 \mid 2 \mid 3}\,; \quad 2\colon \boxed{\times 5 \mid 1}\,; \quad 3\colon \boxed{\times 2 \mid 4}$$

переведут пару $(2^x,\ 1)$ в пару $(2\cdot 5^x,\ 4)$, а приказы

$$4\colon \boxed{:5 \mid 5 \mid 0}\,; \quad 5\colon \boxed{:2 \mid 6 \mid 7}\,; \quad 6\colon \boxed{\times 3 \mid 5}\,;$$

$$7\colon \boxed{:3 \mid 8 \mid 4}\,; \quad 8\colon \boxed{\times 4 \mid 7}$$

переведут $(2\cdot 5^x,\ 4)$ сначала в пару $(2^2\cdot 5^{x-1},\ 4)$, затем в пару $(2^{2\cdot 2}\cdot 5^{x-2},\ 4)$ и через x циклов получим требуемую пару $(2^{2^x},\ 0)$.

С другой стороны, если задана пара $(2^{2^x},\ 1)$, то приказы

$$1\colon \boxed{:4 \mid 2 \mid 7}\,; \quad 2\colon \boxed{\times 20 \mid 3}\,;$$

$$3\colon \boxed{:4 \mid 4 \mid 5}\,; \quad 4\colon \boxed{\times 3 \mid 3}$$

переведут указанную пару в $(3^{2^{x-1}}\cdot 5,\ 5)$, а приказы

5: [:3 | 6 | 1] ; 6: [$\times 2$ | 5]

переведут пару $(3^{2^{x-1}}\cdot 5,\ 5)$ в пару $(2^{2^{x-1}}\cdot 5,\ 1)$. Применяя к последней паре снова весь цикл преобразований 1—6, получим пару $(2^{2^{x-2}}\cdot 5^2,\ 1)$ и т. д. После исчерпания всех четверок мы получим пару $(2\cdot 5^x,\ 7)$, которую приказы

7: [:5 | 8 | 9] ; 8: [$\times 2$ | 7] ; 9: [:2 | 0]

переработают в окончательную пару $(2^x,\ 0)$.

Итак, преобразуя сначала 2^x в 2^{2^x}, затем 2^{2^x} в $2^{2^{f(x)}}$ и, наконец, $2^{2^{f(x)}}$ в $2^{f(x)}$, получим операторный алгоритм с программой вида (4), преобразующий 2^x в $2^{f(x)}$.

Теорема 2 доказана. Указанное в ней кодирование значений x и $f(x)$ в виде 2^x и $2^{f(x)}$ в каком-то смысле неизбежно, так как, например, Я. М. Б а р д з и н ь [5] показал, что невозможен операторный алгоритм с программой вида (4), который бы перерабатывал x в x^2. Поэтому, если мы хотим иметь операторные алгоритмы, перерабатывающие x в $f(x)$ для любой частично рекурсивной функции f, то к списку приказов (4) мы должны присоединить приказы иного вида. В частности, из теоремы 2 непосредственно вытекает, что *для каждой частично рекурсивной функции* $f(x)$ *существует операторный алгоритм, перерабатывающий* x *в* $f(x)$, *программа которого состоит лишь из приказов вида* (4) *и приказов* [2^x | α] , [ex_0 | α] .

В самом деле, согласно теореме 2 существует алгоритм, перерабатывающий 2^x в $2^{f(x)}$ и имеющий программу

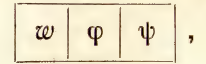
0: [стоп] ; 1: [ ] ; . . . ; s: [ ] ,

состоящую из приказов вида (4). Тогда цепочка приказов

0′: [стоп] ; 1′: [2^x | 1] ; 1: [ ] ; . . . ; s: [ ] ; 0: [ex_0 | 0′]

будет перерабатывать x в $f(x)$.

Понятие операторного алгоритма легко обобщить. Например, можно рассматривать приказы вида



означающие, что, исходя из заданного числа x, следует найти $\varpi(x)$ и затем перейти к выполнению над числом $\varpi(x)$ приказа с номером $\varphi(x)$, если $\varpi(x)$ определено, и перейти к выполнению над x приказа с номером $\psi(x)$, если $\varpi(x)$ не определено. Однако при всех этих обобщениях легко доказывается, что класс вычислимых функций остается совпадающим с классом частично рекурсивных функций.

Весьма интересный класс алгоритмов, перерабатывающих графы, был определен и исследован А. Н. Колмогоровым и В. А. Успенским [45]. Как пишут авторы, эти алгоритмы были получены при попытке воспроизвести в абстрактном определении наибольшее число черт, присущих работе вычислителя, действующего согласно заданной ему программе. Класс функций, вычислимых при помощи алгоритмов Колмогорова — Успенского, оказался снова совпадающим с классом всех частично рекурсивных функций.

Дополнения и примеры

1. В п. 14.2 отмечено, что каждая словарная частично рекурсивная функция $F(\mathfrak{x})$, заданная в алфавите A, вычислима при помощи нормального алгоритма в подходящем расширении алфавита A. Показать, что $F(\mathfrak{x})$ вычислима посредством нормального алгоритма уже в алфавите $\{a_0, A\}$, отличающемся от A лишь одной дополнительной буквой a_0, $a_0 \notin A$ (А. А. Марков [55], Н. М. Нагорный [65]).

2. Невозможно задать нормальный алгоритм в алфавите A, который перерабатывает произвольное слово $\mathfrak{x}$ в алфавите A в его удвоение $\mathfrak{x}\mathfrak{x}$ (Н. М. Нагорный [65]).

3. Какие функции вычислимы операторными алгоритмами, программы которых содержат лишь приказы вида



и стоп-приказ?

4. Согласно теореме 2 из п. 14.2 для каждой частично рекурсивной функции $f(x)$ существует операторный алгоритм, перерабатывающий 2^x в $2^{f(x)}$, причем программа этого алгоритма состоит лишь из приказов указанного в теореме частного вида. Показать, что ни один алгоритм этого типа не может перерабатывать x в x^2 (Я. М. Бардзинь [5]).

5. Показать, что каждая частично рекурсивная функция вычислима посредством операторного алгоритма, программа которого состоит из приказов вида

| +1 | α |

,

| ×c | α |

,

| q | α |

,

| : c | α | β |

,

где $q(x) = x - [\sqrt{x}]^2$, и стоп-приказа.

§ 15. МНОГОЛЕНТОЧНЫЕ МАШИНЫ И ТАГ-СИСТЕМЫ

В отличие от изучавшихся выше машин Тьюринга с одной лентой в данном параграфе будут введены машины Тьюринга, имеющие несколько лент. Легко доказывается, что класс функций, вычислимых на многоленточных машинах, в точности совпадает с классом рекурсивных функций. Это — новое подтверждение тезиса Чёрча. Среди многоленточных машин интересный класс образуют машины, не меняющие состояния ячеек лент. Элементы теории этих машин излагаются в первой половине данного параграфа.

Во второй половине параграфа изучаются свойства еще одного специального класса алгоритмов, введенного Э. Постом под именем систем продукций. Показывается, что даже особым образом определенных *однородных* систем продукций достаточно, чтобы при помощи их можно было вычислять значения любой частично рекурсивной функции.

15.1. Общие многоленточные машины. k-ленточная машина Тьюринга состоит из механического устройства (автомата) A и k лент, разбитых на ячейки (рис. 3). Каждая ячейка может находиться в одном из фиксированного конечного множества состояний. Эти возможные состояния мы условимся обозначать символами $a_0, a_1, \ldots, a_m$, совокупность которых будем называть внешним алфавитом машины. Клетки в состоянии a_0, как и ранее, будут называться *пустыми*. По предположению, автомат A в каждый момент времени находится в одном из конечного множества состояний $q_0, q_1, \ldots, q_n$. Состояния q_1, q_0 называются соответ-

ственно *начальным* и *заключительным*. Машина работает в дискретном времени, так что можно говорить о «состоянии машины», «состоянии автомата», «состояниях ячеек лент» в момент $t = 0$, в момент $t = 1$ и т. д. Предполагается, что в каждый момент времени машина «обозревает» одну ячейку каждой ленты. По определению, состояние (конфигурация) машины считается заданным, если известны состояние q_i, в котором находится автомат A, состояния всех ячеек всех лент и указаны ячейки, воспринимаемые машиной. Таким образом, конфигурация машины вполне описывается системой k машинных слов вида

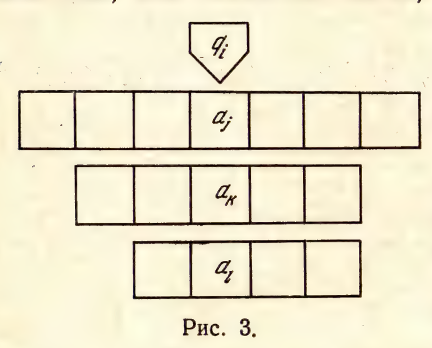


Рис. 3.

$$\begin{gathered} a_{i_{11}} \ldots a_{i_{1s}} q_i a_{i_{1\,s+1}} \ldots a_{i_{1t}}, \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ a_{i_{k1}} \ldots a_{i_{ku}} q_i a_{i_{k\,u+1}} \ldots a_{i_{kv}} \end{gathered} \qquad (1)$$

— по одному слову для каждой ленты.

В зависимости от внутреннего состояния q_i автомата A и состояний $a_{i_{1\,s+1}}, \ldots, a_{i_{k\,u+1}}$ обозреваемых в данный момент ячеек машина $\mathfrak{T}$ переходит в следующую конфигурацию посредством таких действий: а) заменяет состояние каждой воспринимаемой ячейки новым состоянием (которое может совпадать и со старым); б) каждую ленту передвигает на одну ячейку влево, вправо, или оставляет неподвижной; в) переводит внутреннее состояние q_i автомата A в новое состояние q_j. Считается, что если автомат A в какой-то момент времени приходит в заключительное состояние q_0, то машина $\mathfrak{T}$ останавливается и в последующие моменты времени конфигурация машины не меняется.

Говорят, что машина $\mathfrak{T}$ выполняет (или может выполнять) команду

$$q_i a_{\alpha_1} \ldots a_{\alpha_k} \to q_j a_{\beta_1} T_{\gamma_1} \ldots a_{\beta_k} T_{\gamma_k} \quad (\gamma_1, \ldots, \gamma_k = -1, 0, 1), \qquad (2)$$

если, находясь во внутреннем состоянии q_i и воспринимая ячейки в состояниях $a_{\alpha_1}, \ldots, a_{\alpha_k}$, машина $\mathfrak{T}$ заменяет q_i на q_j и указанные состояния состояниями $a_{\beta_1}, \ldots, a_{\beta_k}$, после чего сдвигает ленты соответственно в направлениях $T_{\gamma_1}, \ldots, T_{\gamma_k}$ (T_{-1}, T_1, T_0 означают соответственно сдвиг на одну ячейку влево, вправо и оставление ленты неподвижной). Как и в случае одноленточных машин, предполагается, что в необходимых случаях машина может перед сдвигами надстраивать ленты пустыми ячейками.

Задать программу машины $\mathfrak{T}$ — это значит для каждого слова

$$q_i a_{\alpha_1} \ldots a_{\alpha_k} \qquad (i = 1, \ldots, n; \quad \alpha_1, \ldots, \alpha_k = 0, 1, \ldots, m)$$

задать какую-нибудь команду вида (2).

Ясно, что, зная программу машины $\mathfrak{T}$, мы можем путем серии указанных операций для любой конфигурации (1) машины $\mathfrak{T}$ в заданный момент времени вычислить ее конфигурацию в любой из последующих моментов времени. Иначе говоря, каждая k-ленточная машина Тьюринга определяет алгоритм для переработки систем машинных слов вида (1).

Пусть $f(x_1, \ldots, x_s)$ — какая-нибудь частичная функция от s переменных. Говорят, что f нормально вычислима на $(s+k)$-ленточной машине Тьюринга $\mathfrak{T}$ с внешним алфавитом 0, 1, $a_2, \ldots, a_m$, если для любых натуральных $n_1, \ldots, n_s$ машина $\mathfrak{T}$, начав работать в конфигурации

$$10^{n_1-1}q_10, \ldots, 10^{n_s-1}q_10, q_11, \ldots, q_11 \qquad (10^{-1}q\mathfrak{a} = q1\mathfrak{a}),$$

по истечении конечного времени перейдет в конфигурацию

$$10^{f(n_1, \ldots, n_s)-1}q_00 \ldots 0, q_0 10 \ldots 0, \ldots, q_010 \ldots 0$$

при условии, что $f(n_1, \ldots, n_s)$ определено, и будет работать вечно, то есть не переходя во внутреннее состояние q_0, в противном случае. При этом, как и в случае одноленточных машин, можно предполагать, что в продолжение всей работы машина не надстраивает ячеек слева ни на одной ленте.

Теорема 1. *Для того чтобы частичная функция $f(x_1, \ldots, x_s)$ была вычислима на $(s+k)$-ленточной машине Тьюринга с внешним алфавитом* 0, 1, $a_2, \ldots, a_m$

(*m — любое натуральное число*), *необходимо и достаточно, чтобы f была частично рекурсивной.*

Мы не будем излагать подробное доказательство этой теоремы. Заметим лишь, что вычислимость на $(s+k)$-ленточной машине частично рекурсивных функций есть прямое следствие вычислимости этих функций на одноленточной машине, а обратное утверждение есть непосредственное следствие теоремы о частичной рекурсивности функций, вычислимых при помощи нормальных алгоритмов (п. 14.2).

15.2. Машины Минского. Способность менять состояния ячеек ленты — существенная черта машин Тьюринга. Однако для многоленточных машин эта способность перестает быть необходимой. А именно, Лембеком [47] было показано, что каждая частично рекурсивная функция вычислима на машине с достаточным числом лент так, что машина не меняет состояния ячеек лент и не надстраивает левых ячеек. Почти одновременно Минский [60] показал, что это же могут делать и двухленточные машины, не изменяющие состояния ячеек и не надстраивающие ячеек слева, при условии, что значения аргумента и функции кодируются степенями двойки так, как это делалось в теореме 2 из п. 14.3.

Многоленточные машины Тьюринга, у которых ленты слева не надстраиваются, все ячейки лент, за исключением самых левых, всегда пусты, а состояния самых левых ячеек постоянны, мы будем называть машинами Минского. Внешний алфавит этих машин предполагается состоящим из символов 0, 1, причем 0 — символ пустого состояния, а все самые левые клетки находятся в состоянии 1.

Для полного описания машины Минского достаточно указать совокупность всех ее внутренних состояний и программу машины, то есть совокупность команд вида

$$q_i a_1 \ldots a_s \to q_\alpha T_{\alpha_1} \ldots T_{\alpha_s}$$

$$(i=1, \ldots, n;\ a_\lambda = 0,\ 1;\quad \alpha = 0,\ 1, \ldots, n;\ \alpha_\lambda = 0,\ 1, -1), \tag{1}$$

которые способна выполнять данная машина. Так как 1 есть состояние самой левой ячейки, то в командах (1) из $a_\lambda = 1$ должно следовать неравенство

$$\alpha_\lambda \neq -1. \tag{2}$$

Этим условием и ограничивается произвол в задании команд (1).

Т е о р е м а 1. *Для каждой частично рекурсивной функции $f(x)$ существует трехленточная машина Минского, вычисляющая эту функцию, то есть переходящая из конфигурации* $10^{x-1}q_10$, q_11, q_11 *в конфигурацию* $10^{f(x)-1}q_00$, q_01, q_01, *если $f(x)$ определено, и работающая вечно, если $f(x)$ не определено.*

Условимся изображать четверкой чисел $(a, b, c; i)$ ту конфигурацию трехленточной машины Минского, которая характеризуется машинными словами $10^{a-1}q_i0$, $10^{b-1}q_i0$, $10^{c-1}q_i0$. Напомним, что $10^{-1}q_i0$ есть слово q_i10. Таким образом, числа a, b, c равняются просто длинам концов лент, находящихся слева от воспринимаемых машиной ячеек, а i — номер (или символ) внутреннего состояния машины.

Согласно теореме 2 из п. 14.3 существует операторный алгоритм $\boldsymbol{A}$, перерабатывающий 2^x в $2^{f(x)}$ и имеющий программу вида

$$0: \boxed{\text{стоп}}\ ;\ 1: \boxed{\zeta_1 \mid \alpha_1 \mid \beta_1}\ ;\ \ldots,\ n: \boxed{\zeta_n \mid \alpha_n \mid \beta_n}\ , \qquad (2)$$

где ζ_i означает одну из операций $\times 2$, $\times 3$, $\times 5$, $:30$. Все эти операции перерабатывают числа вида $2^a3^b5^c$ в числа того же вида. Поэтому, начиная перерабатывать число 2^x по программе (2), мы после k-го шага вычислений получим (см. п. 14.3) пару вида $(2^a\cdot 3^b\cdot 5^c, i)$, где $2^a\cdot 3^b\cdot 5^c$ — полученное в этот момент число, а i — номер приказа в программе (2), который следует далее выполнять над числом $2^a\cdot 3^b\cdot 5^c$.

С парой $(2^a\cdot 3^b\cdot 5^c, i)$ сопоставляем конфигурацию $(a, b, c; i)$ трехленточной машины Минского, имеющей внутренние состояния $q_0, q_1, \ldots, q_n$. Нам остается подобрать программу (1) машины M так, чтобы выполнялось условие: *если алгоритм A переводит пару $(2^a\cdot 3^b\cdot 5^c, i)$ в пару $(2^u\cdot 3^v\cdot 5^w, j)$, то машина M должна переходить из конфигурации $(a, b, c; i)$ в конфигурацию $(u, v, w; j)$.*

Эта цель, очевидно, будет достигнута, если мы поступим следующим образом:

а) Если $\zeta_i = :30$, то вносим в программу M команды

$$q_i 000 \to q_{\alpha_i} T_1 T_1 T_1,$$

$$q_i abc \to q_{\beta_i} T_0 T_0 T_0 \quad (a+b+c \neq 0).$$

б) Если $\zeta_i = \times 2$, то в программу M вносим команды

$$q_i abc \to q_{\alpha_i} T_{-1} T_0 T_0 \quad (a,\ b,\ c = 0,\ 1).$$

в Если $\zeta_i = \times 3$, то вносим команды

$$q_i abc \to q_{\alpha_i} T_0 T_{-1} T_0.$$

г) Если $\zeta_i = \times 5$, то вносим команды

$$q_i abc \to q_{\alpha_i} T_0 T_0 T_{-1}.$$

Так как алгоритм $\boldsymbol{A}$ перерабатывает пару $(2^x,\ 1)$ в пару $(2^{f(x)},\ 0)$, то машина M с командами а) — г) из конфигурации $(x,\ 0,\ 0;\ 1)$ переходит в конфигурацию $(f(x),\ 0,\ 0;\ 0)$, что и требовалось.

Рассмотрим теперь более детально *двухленточные* машины Минского. Пусть $q_0,\ q_1,\ \ldots,\ q_n$ — внутренние состояния какой-нибудь двухленточной машины Минского M. Тогда для полного задания машины M достаточно задать ее программу, состоящую из команд вида

$$q_i ab \to q_\alpha T_\beta T_\gamma,$$

удовлетворяющих условию (2).

Далеко не все частично рекурсивные функции вычислимы на двухленточных машинах Минского в указанном выше смысле. Например, не существует двухленточной машины Минского, перерабатывающей x в x^2 (см. Я. М. Б а р д з и н ь [5]). Тем не менее, если значения аргумента и функции кодировать степенями двойки, то вычислимость частично рекурсивных функций восстанавливается.

Т е о р е м а 2 (М и н с к и й [60]). *Для каждой частично рекурсивной функции $f(x)$ существует двухленточная машина Минского, которая для любого натурального x перерабатывает число 2^x в число $2^{f(x)}$, если $f(x)$ определено, и работает безостановочно, не переходя в заключительное внутреннее состояние q_0, если $f(x)$ не определено.*

Для краткости условимся тройкой $(a,\ b;\ q_i)$ обозначать ту конфигурацию машины Минского, при которой q_i — внутреннее состояние машины, а a, b — расстояния от воспринимаемых машиной ячеек до левых концов первой и соответственно второй лент. Например, $(0,\ 2;\ q_i)$ есть конфигурация, определенная машинными словами q_i1, $10q_i0$ и изображенная на рис. 4. Формулой $(a,\ b;\ q_i) \vdash (c,\ d;\ q_j)$ будем обозначать утверждение, что машина, начав работать в конфигурации $(a,\ b;\ q_i)$ через конечное число тактов работы приобретет конфигурацию $(c,\ d;\ q_j)$. Если машина из конфигурации $(a,\ b;\ q_i)$ непосредственно переходит в конфигурацию $(c,\ d;\ q_j)$, то пишем

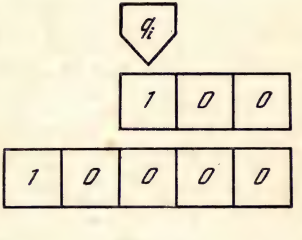


Рис. 4.

$$(a,\ b;\ q_i) \to (c,\ d;\ q_j). \tag{3}$$

Как уже говорилось, задать машину Минского — это значит указать ее программу. Однако вместо того, чтобы указывать команды, составляющие программу машины, мы будем писать преобразования (3), выполняемые машиной. При этом следует иметь в виду такие обстоятельства. Пусть, например, указывается, что машина выполняет преобразование

$$(2,\ 5;\ q_i) \to (1,\ 6;\ q_j). \tag{4}$$

Отсюда следует, что машина выполняет команду

$$q_i00 \to q_jT_1T_{-1}$$

и потому для любых положительных a, b имеем

$$(a,\ b;\ q_i) \to (a-1,\ b+1;\ q_j). \tag{5}$$

В то же время наличие в программе машины преобразования (4) еще ничего не говорит о том, во что перейдут конфигурация $(0,\ b;\ q_i)$ или конфигурации $(b,\ 0;\ q_i)$, $(0,\ 0;\ q_i)$, где $b>0$. Поэтому, присоединив к программе машины преобразование (4), мы не можем присоединять к программе преобразования, противоречащие преобразованиям (5), но

все еще можем по произволу присоединить преобразования

$$(0,\ b;\ q_i) \to (u,\ v;\ q_\alpha) \qquad (u=0,\ 1;\ v=b,\ b\pm 1),$$
$$(b,\ 0;\ q_i) \to (u_1,\ v_1;\ q_\beta) \qquad (v_1=0,\ 1;\ u_1=b,\ b\pm 1),$$
$$(0,\ 0;\ q_i) \to (u_2,\ v_2;\ q_\gamma) \qquad (u_2,\ v_2=0,\ 1).$$

Нам надо для каждой частично рекурсивной функции $f(x)$ указать программу машины Минского, перерабатывающей 2^x в $2^{f(x)}$. Нижеследующая лемма сводит эту задачу к двум простым частным случаям.

Лемма 1. *Пусть операторный алгоритм* **A** *с программой*

$$0: \boxed{\text{стоп}}\ ;\quad 1: \boxed{\zeta_1 \,|\, \alpha_1 \,|\, \beta_1}\ ;\ \ldots;\ s: \boxed{\zeta_s \,|\, \alpha_s \,|\, \beta_s} \qquad (6)$$

вычисляет функцию $g(x)$ *и пусть для некоторой машины Минского* M *с внутренними состояниями* $q_0,\ q_1,\ \ldots,\ q_s,\ q_{1i},\ \ldots,\ q_{si}$ $(i=1,\ 2,\ \ldots)$ *имеем*

$$(z,\ 0;\ q_i) \vdash (\zeta_i(z),\ 0;\ q_{\alpha_i}),\ \textit{если}\ \zeta_i(z)\ \textit{определенное}, \qquad (7)$$

$$(z,\ 0;\ q_i) \vdash (z,\ 0;\ q_{\beta_i}),\ \textit{если}\ \zeta_i(z)\ \textit{неопределенное}. \qquad (8)$$

Тогда машина M *вычисляет ту же функцию* $g(x)$, *что и алгоритм* **A**.

В самом деле, пусть, начиная с заданного числа x, алгоритм A после m шагов дает число x_m и указание выполнять приказ с номером i. Покажем, что в этом случае

$$(x,\ 0;\ q_1) \vdash (x_m,\ 0;\ i).$$

Для $m=0$ это утверждение тривиально. Пусть оно верно для какого-нибудь m. Если $\zeta_i(x_m)$ определено, то алгоритм A после $(m+1)$-го шага дает число $\zeta_i(x_m)$ и указание выполнять приказ с номером α_i. Но в этом случае

$$(x_m,\ 0;\ q_i) \vdash (\zeta_i(x_m),\ 0;\ q_{\alpha_i})$$

и потому

$$(x,\ 0;\ q_1) \vdash (\zeta_i(x_m),\ 0;\ q_{\alpha_i}).$$

Аналогично доказывается это соотношение и в случае, когда $\zeta_i(x_m)$ не определено.

Пусть теперь алгоритм $\boldsymbol{A}$ перерабатывает число x в число y. Это значит, что через какое-то число шагов m алгоритм $\boldsymbol{A}$ дает число y и указание выполнять приказ с номером 0. По доказанному

$$(x,\ 0;\ q_1) \vdash (y,\ 0;\ q_0),$$

то есть машина M перерабатывает x в y.

Если, начиная с x, алгоритм $\boldsymbol{A}$ работает вечно, то получаем бесконечную последовательность

$$(x,\ 0;\ q_1) \vdash (x_1,\ 0;\ q_{i_1}) \vdash \ldots,$$

показывающую, что машина M в этом случае также работает вечно.

Перейдем к доказательству основной теоремы. Пусть $f(x)$ — заданная частично рекурсивная функция. Согласно теореме 2 из п. 14.3 существует операторный алгоритм $\boldsymbol{A}$, перерабатывающий 2^x в $2^{f(x)}$, программа которого имеет вид (6), где i-й приказ есть или

$$i\colon \boxed{\times c \mid \alpha} \qquad (c=2,\ 3,\ 5), \tag{9}$$

или

$$i\colon \boxed{:c \mid \alpha \mid \beta} \qquad (c=30). \tag{10}$$

Пусть M — машина Минского с внутренними состояниями $q_0,\ q_1,\ \ldots,\ q_n$, каждое из которых соответствует определенному приказу программы (6), и дополнительными состояниями q_{ij} ($i = 1,\ \ldots;\ j = 0,\ 1,\ \ldots$), точное число которых будет указано ниже. Нам надо найти такую программу для машины M, чтобы были выполнены условия (7), (8) леммы 1. Рассмотрим какое-нибудь i, $1 \leqslant i \leqslant s$. Пусть i-й приказ в программе (6) имеет вид (9). Постараемся написать такую серию команд для машины M, содержащих лишь внутренние состояния $q_i,\ q_{i0},\ \ldots,\ q_{ic},\ q_\alpha$, в результате которой условие (7) для этого i будет выполнено. Пусть $a > 0$. Вносим в программу машины M следующую серию преобразований:

$$(a,\ 0;\ q_i) \to (a,\ 1;\ q_{i1}) \to$$
$$\to (a,\ 2;\ q_{i2}) \to \ldots \to (a,\ c;\ q_{ic}) \to (a-1,\ c;\ q_i) \tag{11}$$

и дополняем ее еще преобразованием

$$(a,\ 1;\ q_i) \to (a,\ 2;\ q_{i1}). \qquad (12)$$

Тогда, если в серии (11) $a - 1 > 0$, то в силу команд, отвечающих преобразованиям (11), (12), конфигурация $(a - 1,\ c;\ q_i)$ далее будет меняться следующим образом:

$$(a-1,\ c;\ q_i) \to (a-1,\ c+1;\ q_{i1}) \to \ldots \to$$
$$\to (a-1,\ 2c;\ q_{ic}) \to (a-2,\ 2c;\ q_i) \to (a-2,\ 2c+1;\ q_{i1}) \to$$
$$\to \ldots \to (0,\ ac;\ q_i).$$

Чтобы переставить числа 0 и ac, дополняем программу преобразованиями

$$(0,\ ac;\ q_i) \to (1,\ ac-1;\ q_{i0}) \to (2,\ ac-2;\ q_{i0}) \to$$
$$\to \ldots \to (ac,\ 0;\ q_{i0}) \to (ac,\ 0;\ q_\alpha). \qquad (13)$$

В результате условий (11), (12) и (13) для машины M будет гарантировано свойство

$$(a,\ 0;\ q_i) \vdash (ca,\ 0;\ q_\alpha) \qquad (a > 0). \qquad (14)$$

Чтобы гарантировать свойство (14) и для $a = 0$, внесем в программу M преобразование

$$(0,\ 0;\ q_i) \to (0,\ 0;\ q_\alpha). \qquad (15)$$

Итак, внося в программу машины M команды (11), (12), (13), (15) для тех i, для которых i-й приказ программы $\boldsymbol{A}$ имеет вид (9), мы обеспечим для этих значений i выполнение условия (7) из леммы 1.

Теперь рассмотрим более сложный случай, когда i-й приказ имеет вид (10). Прежде всего в программу машины M вносим команды, отвечающие для $x \geqslant c$, $y \geqslant 0$ следующей серии преобразований:

$$(x,\ y;\ q_i) \to (x-1,\ y+1;\ q_{i1}) \to (x-2,\ y+1;\ q_{i2}) \to$$
$$\to \ldots \to (x-c,\ y+1;\ q_{ic}) \to (x-c,\ y+1;\ q_i), \qquad (16)$$

и команды, отвечающие преобразованиям (13) и (14). Вследствие команд (16) для $x \geqslant c$, $y \geqslant 0$ имеем

$$(x,\ y;\ q_i) \vdash (x-c,\ y+1;\ q_i)$$

и, значит,

$$(x,\ 0;\ q_i) \vdash (0,\ x:c;\ q_i),$$

если x делится на c. Так как из (16), (13), (14) вытекает

$$(0,\ x:c;\ q_i) \vdash (x:c,\ 0;\ q_\alpha),$$

то

$$(x,\ 0;\ q_i) \vdash (x:c,\ 0;\ q_\alpha),\ \text{если}\ x:c\ \text{определено}.$$

Пусть теперь x не делится на c, $x = cz + r$, $0 < r < c$. В силу (16) получим

$$(x,\ 0;\ q_i) \vdash (r,\ z;\ q_i) \vdash (0,\ z+1;\ q_{ir}).$$

Нам надо теперь от конфигурации $(0,\ z+1;\ q_{ir})$ перейти к конфигурации $(x,\ 0;\ q_\beta)$. Поэтому к программе M добавляем преобразования ($z = 1, 2, \ldots$; $r = 1, \ldots, c-1$):

$$(0,\ z+1;\ q_{ir}) \to (1,\ z;\ q_{ir1}) \to \ldots \to (c,\ z;\ q_{irc}) \to \\ \to (c+1,\ z-1;\ q_{ir1}), \quad (17)$$

в силу которых будем иметь

$$(0,\ z+1;\ q_{ir}) \vdash (cz+1,\ 0;\ q_{ir1}).$$

Наконец, добавляя в программу преобразования

$$(cz+1,\ 0;\ q_{ir1}) \to (cz+2,\ 0;\ q_{ir2}) \to \ldots \\ \ldots \to (cz+r,\ 0;\ q_{irr}) \to (cz+r,\ 0;\ q_\beta), \quad (18)$$

получим

$$(x,\ 0;\ q_i) \vdash (x,\ 0;\ q_\beta),\ \text{если}\ x:c\ \text{не определено}.$$

Мы видим, что машина Минского с программой, построенной согласно правилам (11) — (13), (15) — (18), удовлетворяет условиям леммы 1, что и требовалось.

Так же, как и для одноленточных машин Тьюринга (п. 12.5), для машин Минского естественно формулируются следующие две проблемы.

1) Проблема остановки. *Задана машина Минского M (своей программой). Указать алгоритм, распознающий те конфигурации $(a,\ b;\ q_1)$, начиная с которых машина останавливается через конечное число тактов работы.*

2) Проблема конфигураций. *Дана машина Минского M. Указать алгоритм, позволяющий узнать для любой заданной конфигурации $(x,\ 0;\ q_0)$, существует ли*

начальная конфигурация $(a,\ 0;\ q_1)$, *которая через конечное число тактов работы переходит в конфигурацию* $(x,\ 0;\ q_0)$.

Обе проблемы решаются отрицательно. В самом деле, пусть $f(x)$ — частично рекурсивная функция, область определенности которой не рекурсивна. Строим двухленточную машину Минского M, перерабатывающую 2^x в $2^{f(x)}$. Если бы существовал алгоритм, решающий проблему остановки машины M, то этот алгоритм решал бы и проблему вхождения числа x в область определенности функции $f(x)$, так как x входит в область определенности f тогда и только тогда, когда машина M, начиная работать в конфигурации $(2^x,\ 0;\ q_1)$, останавливается через конечное число тактов работы.

Аналогично, пусть $f(x)$ — примитивно рекурсивная функция, совокупность значений которой не рекурсивна. Строим двухленточную машину Минского M, перерабатывающую 2^x в $2^{f(x)}$. Как и выше, легко убеждаемся, что для M не разрешима проблема конфигураций.

При доказательстве теоремы 2 мы пользовались теоремой 2 из п. 14.3 о том, что для каждой частично рекурсивной функции $f(x)$ существует операторный алгоритм, перерабатывающий 2^x в $2^{f(x)}$, программа которого состоит лишь из приказов вида

:30	α	β

,

×c	α

$(c = 2,\ 3,\ 5)$.

В свою очередь, с помощью теоремы 2 можно еще сократить этот список приказов, правда, усложнив кодирование. Покажем, что имеет место

Т е о р е м а 3 (М и н с к и й [60]). *Для каждой частично рекурсивной функции* $f(x)$ *существует операторный алгоритм, перерабатывающий* 2^{2^x} *в* $2^{2^{f(x)}}$, *программа которого состоит лишь из приказов вида*

×2	α

,

×3	α

,

:6	α	β

.

Пусть M — двухленточная машина Минского с внутренними состояниями $q_0,\ q_1,\ \ldots,\ q_n$, переходящая из конфигурации $(2^x,\ 0;\ q_1)$ в конфигурацию $(2^{f(x)},\ 0;\ q_0)$, если $f(x)$ определено, и работающая вечно в противном

случае. Машина M может двигать одновременно или обе ленты, или только одну. Покажем сначала, что M можно заменить такой машиной M', которая при каждом такте работы непременно либо двигает обе ленты на 1 шаг вправо, либо двигает только одну ленту (1-ю или 2-ю) на 1 шаг влево. Действительно, пусть (1) — программа машины M. Если эта программа содержит приказ

$$q_i00 \rightarrow q_\alpha T_1 T_{-1}, \tag{19}$$

то вводим дополнительные внутренние состояния q_{i1}, q_{i2} и вместо приказа (19) пишем приказы

$$\begin{aligned} &q_i00 \rightarrow q_{i1}T_1T_1, \\ &q_{i1}ab \rightarrow q_{i2}T_0T_{-1} \quad (a,\ b=0,\ 1), \\ &q_{i2}ab \rightarrow q_\alpha T_0 T_{-1}. \end{aligned} \tag{20}$$

Аналогично поступаем и с другими приказами из программы (1). Ясно, что новая машина M' с внутренними состояниями q_i, q_{ij} и приказами вида (20) будет по-прежнему из конфигурации $(2^x,\ 0;\ q_1)$ переходить в конфигурацию $(2^{f(x)},\ 0;\ q_0)$ и в то же время будет сдвигать ленты в соответствии с наложенными требованиями. Обозначим внутренние состояния машины M' через $q_0, q_1, \ldots, q_n, q_{n+1}, \ldots, q_s$. Программа машины M' — это совокупность команд вида

$$\begin{aligned} &q_i00 \rightarrow q_{\alpha_i}R_{\varkappa_i} \quad (\varkappa_i=0,\ -1,\ 1), \\ &q_i10 \rightarrow q_{\beta_i}R_{\lambda_i}, \\ &q_i01 \rightarrow q_{\gamma_i}R_{\mu_i} \quad (\lambda_i,\ \mu_i,\ \nu_i=0,\ 1;\ i=1,\ \ldots,\ s), \\ &q_i11 \rightarrow q_{\delta_i}R_{\nu_i}, \end{aligned} \tag{21}$$

где R_{-1} означает сдвиг обеих лент вправо, R_0 — сдвиг влево 1-й ленты, R_1 — сдвиг влево 2-й ленты.

Условимся конфигурацию $(a,\ b;\ q_i)$ изображать парой $(2^a \cdot 3^b,\ q_i)$. Тогда сдвиг обеих лент вправо будет влечь за собой деление первого числа пары на 6, а сдвиг одной из лент влево будет влечь умножение этого числа на 2 или 3. Исходя из программы (21), легко сразу написать и программу операторного алгоритма **A**, который число $2^a \cdot 3^b$ и указание i перерабатывает в число $2^c \cdot 3^d$ и указание j,

если машина M' из конфигурации $(2^a \cdot 3^b,\ q_i)$ переходит непосредственно в конфигурацию $(2^c \cdot 3^d,\ q_j)$. Для этого часть приказов алгоритма **А** занумеруем числами 0, 1, ..., s, а часть — парами чисел ij и далее поступим следующим способом. Берем для какого-нибудь i четверку команд (21) из программы машины M' и пусть $\varkappa_i = -1$. Чтобы из числа $2^a \cdot 3^b$ с помощью команд (21) получить число $2^c \cdot 3^d$, надо узнать, нет ли среди чисел a, b нулей. Для этого делим $2^a \cdot 3^b$ на 6. Если делится без остатка, то согласно первой команде четверки (21) частное и будет требуемым числом. Если $2^a \cdot 3^b$ на 6 не делится, то среди чисел a, b есть равное 0 и потому надо воспользоваться одной из последующих команд системы (21). Итак, в программу алгоритма **А** вносим приказ

$$i: \boxed{:6 \mid \alpha_i \mid i.0}\,, \qquad (22)$$

который дает требуемую пару $(2^c \cdot 3^d,\ q_{\alpha_i})$, если $2^a \cdot 3^b$ делится на 6, и отсылает к приказу $i.0$, если $2^a \cdot 3^b$ на 6 не делится. Если в последнем случае число $2^a \cdot 3^b \cdot 3$ делится на 6, то $a > 0$, $b = 0$ и мы должны воспользоваться второй из команд (21). В соответствии с этим в программу **А** вносим приказы

$$\begin{array}{ll} i.0: \boxed{\times 3 \mid i.1}\,; & i.1: \boxed{:6 \mid i.2 \mid i.4}\,; \\ i.2: \boxed{\times 2 \mid i.3}\,; & i.3: \boxed{\times p_{\lambda_i} \mid \beta_i}\ (p_0 = 2,\ p_1 = 3), \end{array} \qquad (23)$$

дающие искомую пару $(2^c \cdot 3^d,\ q_{\beta_i})$, если $a > 0$, и дающие пару $(2^a \cdot 3^{b+1},\ i.4)$ в противном случае. Приказы

$$i.4: \boxed{\times 2 \mid i.5}\,; \quad i.5: \boxed{:6 \mid i.6} \qquad (24)$$

переводят пару $(2^a \cdot 3^{b+1},\ i.4)$ в пару $(2^a \cdot 3^b,\ i.6)$, где $a = 0$. Если $b > 0$, то должны воспользоваться 3-й из команд (21)

и потому в программу **A** добавляем приказы

$$i.6: \boxed{\times 2 \mid i.7}\,; \quad i.7: \boxed{:6 \mid i.8 \mid i.10}\,;$$
$$i.8 \boxed{\times 3 \mid i.9}\,; \quad i.9: \boxed{\times p_{\mu_i} \mid \gamma_i}\,, \tag{25}$$

приводящие при $b > 0$ к паре $(2^c \cdot 3^d,\ q_j)$ и при $b = 0$ к паре $(2^{a+1},\ i.10)$. Наконец, приказы

$$i.10: \boxed{\times 3 \mid i.11}\,; \quad i.11: \boxed{:6 \mid i.12}\,; \quad i.12: \boxed{\times p_{\nu_i} \mid \delta_i} \tag{26}$$

и для случая $a = b = 0$ дадут требуемую пару $(2^c \cdot 3^d,\ q_j)$.

Аналогичным путем пишутся приказы (22) — (26) и для $\varkappa_i = 0,\ 1$. Объединяя приказы, построенные указанным образом для $i = 1, 2, \ldots, s$, и присоединяя к ним начальный стоп-приказ, получим программу операторного алгоритма **A**, имеющую требуемый теоремой 3 вид. По условию машина M из конфигурации $(2^x,\ 0;\ q_1)$ преобразуется в конфигурацию $(2^{f(x)};\ 0;\ q_0)$ или в новом кодировании из конфигурации $(2^{2^x},\ q_1)$ преобразуется в конфигурацию $(2^{2^{f(x)}},\ q_0)$. Отсюда следует, что алгоритм **A** с программой, построенной по правилам (22) — (26), перерабатывает число 2^{2^x} в число $2^{2^{f(x)}}$.

15.3. Однородные продукции. ТАГ-системы. В п. 14.1 было введено понятие системы продукций, определяемой алфавитом $C = \{c_1,\ c_2,\ \ldots,\ c_m\}$ и системой базисных продукций

$$\mathfrak{a}_i W \to W \mathfrak{b}_i \quad (i = 1\ \ldots,\ s). \tag{P}$$

Чтобы при помощи этой системы P определить понятие P-вычислимых частичных числовых функций, надо каким-то образом кодировать числа в алфавите C. Мы примем такое соглашение. Пусть алфавит C системы P содержит символы a_0, a_1, A_0, A_1, B_0, B_1 и произвольное число других символов. Фиксируем, кроме того, еще произвольное натуральное положительное число w и будем произвольное натуральное число x кодировать словом $A_1 a_1^x B_1^{w-1}$. Мы будем говорить, что частичная числовая функция $f(x)$ вычисляется системой продукций P, если система P перерабатывает слово $A_1 a_1^x B_1^{w-1}$ в слово $A_0 a_0^{f(x)} B_1^{w-1}$ для тех x, для которых $f(x)$ определено, и система P без конца преобразует слово $A_1 a_1^x B_1^{w-1}$ для тех x, для которых $f(x)$ не определено.

Из результатов п. 14.2 непосредственно следует, что каждая частичная числовая функция, вычислимая какой-либо системой продукций, необходимо частично рекурсивна. Из тех же результатов нетрудно было бы вывести и обратное утверждение. Однако вместо этого мы хотим здесь доказать более тонкую теорему, касающуюся вычислимости числовых функций при помощи систем продукций, имеющих специальный вид.

По определению, *однородная* система продукций (или ТАГ-система) T задается своим алфавитом $C = \{a_1, a_2, \ldots, a_m\}$, положительным натуральным числом w, называющимся *шагом* системы, и особыми элементарными преобразованиями

$$a_1 \xrightarrow{w} \mathfrak{b}_1, \quad a_2 \xrightarrow{w} \mathfrak{b}_2, \ \ldots, \ a_m \xrightarrow{w} \mathfrak{b}_m, \tag{T}$$

где $\mathfrak{b}_1, \ldots, \mathfrak{b}_m$ — какие-нибудь слова в алфавите C.

T-продукцией над каким-нибудь словом $\mathfrak{x}$, записанным в алфавите C, называется следующий процесс: в последовательности (T) ищем слово $\mathfrak{b}_i$, отвечающее начальной букве a_i слова $\mathfrak{x}$. Затем отбрасываем в $\mathfrak{x}$ первые w букв и к остатку справа приписываем слово $\mathfrak{b}_i$. Если длина слова $\mathfrak{x}$ меньше w, то T-продукция называется невыполнимой над $\mathfrak{x}$ или неприменимой к $\mathfrak{x}$.

Однородные системы продукций можно рассматривать как обычные системы продукций вида (P), подчиненные требованиям:

1) все начальные слова $\mathfrak{a}_1, \ldots, \mathfrak{a}_s$ системы (P) имеют одну и ту же длину w;

2) слова $\mathfrak{b}_1, \ldots, \mathfrak{b}_s$ в продукциях (P) зависят лишь от начальных букв слов $\mathfrak{a}_1, \ldots, \mathfrak{a}_s$;

3) среди слов $\mathfrak{a}_1, \ldots, \mathfrak{a}_s$ системы (P) содержится любое слово длины w над заданным алфавитом.

Чтобы однородную систему продукций, заданную шагом w и элементарными преобразованиями (T), записать в виде системы (P) обычных продукций, достаточно каждое преобразование $a_i \to \mathfrak{b}_i$ в цепочке (T) заменить последовательностью всевозможных продукций вида $a_i \mathfrak{a} W \to W \mathfrak{b}_i$, где $\mathfrak{a}$ — произвольное слово длины $w - 1$ в заданном алфавите.

В однородной системе продукций с шагом w слова, к которым продукции не применимы, имеют длину, меньшую w. Чтобы избавиться от этого недостатка, введем еще понятие однородной системы продукций с заключительной буквой.

Система продукций в алфавите $a_0, a_1, \ldots, a_m$ называется *однородной системой продукций шага w с заключительной буквой a_0*, если совокупность ее продукций можно представить в виде

$$a_1 \xrightarrow{w} \mathfrak{b}_1, \quad a_2 \xrightarrow{w} \mathfrak{b}_2, \ \ldots, \ a_m \xrightarrow{w} \mathfrak{b}_m \tag{T$_0$}$$

(буква a_0 слева не встречается), где $\mathfrak{b}_1, \ldots, \mathfrak{b}_m$ — непустые слова в алфавите $a_0, a_1, \ldots, a_m$.

В отличие от однородной системы (T) в системе (T_0) продукциям не поддаются не только слова длины, меньшей w, но и все слова, начинающиеся заключительной буквой a_0.

Теорема 1 (Минский [60]). *Для каждой частично рекурсивной функции $f(x)$ существует однородная система продукций с заключительной буквой A_0 и шагом w, для каждого x перерабатывающая слово $A_1 a_1^{2x} B_1^{w-1}$ в слово $A_0 a_0^{2^{f(x)}} B_0^{w-1}$ при $f(x)$ определенном и работающая вечно при $f(x)$ неопределенном.*

Для каждой ограниченной частично рекурсивной функции $f(x)$ существует однородная система продукций (без заключительной буквы) шага $w > 2^{f(x)}$, перерабатывающая слово $A_1 a_1^{2x} B_1^{w-1}$ в слово $a_0^{2^{f(x)}}$.

Согласно п. 14.3 существует операторный алгоритм, перерабатывающий 2^x в $2^{f(x)}$ и обладающий программой вида

$$0: \boxed{\text{стоп}}\ ; \quad 1: \boxed{\zeta_1 \mid \alpha_1 \mid \beta_1}\ ; \ \ldots; \quad s: \boxed{\zeta_s \mid \alpha_s \mid \beta_s}\ , \tag{1}$$

где ζ_i имеет одно из значений $\times 2$, $\times 3$, $\times 5$, $:30$. Число $w = 30$ мы примем в качестве шага искомой системы продукций. Алфавит этой системы будет состоять из основных букв A_0, a_0, A_1, a_1, ... , A_s, a_s, попарно отвечающих приказам (1), и ряда других букв b_i, b_{i0}, B_i, B_{ij}, C_i ($i, j = 0, 1, \ldots$).

Процесс переработки заданного числа 2^x посредством алгоритма (1) распадается на шаги так, что после какого-то k-го шага получается число z (промежуточный результат) и номер i приказа, который должен быть выполнен над числом z при следующем шаге вычислений. Если $i = 0$, то процесс вычислений обрывается и z — результат переработки 2^x посредством алгоритма (1). С парой чисел (z, i) мы сопоставим слово $A_i a_i^z B_i^{w-1}$ и постараемся построить систему продукций T так, чтобы эта система преобразовывала слово $A_i a_i^z B_i^{w-1}$ в слово $A_j a_j^u B_j^{w-1}$, если алгоритм (1) преобразует пару (z, i) в пару (u, j).

Для каждого i ($1 \leqslant i \leqslant s$) может быть 2 случая: $\zeta_i = \times c$, $\zeta_i = :w$. Рассмотрим их отдельно.

1-й случай: $i: \boxed{\times c \mid \alpha}$.

Пишем следующие однородные продукции:

$$A_i \to B_{i0} \ldots B_{iw},\ a_i \to b_i^{w^2},\ B_i \to b_i^w,\ B_{i0} \to C_i^w A_\alpha,$$

$$b_i \to a_\alpha^c,\ C_i \to B_\alpha^{w-1},\ B_{i\,w-r} \to b_i^{r(w-1)} C_i^w A_\alpha \ (r = 1, \ldots, w-1) \tag{2}$$

Посмотрим, во что переведут эти продукции слово $A_i a_i^z B_i^{w-1}$. Представляя z в виде $z = lw + r$, $0 \leqslant r < w$, и предполагая, что

$lr \neq 0$, мы из (2) непосредственно получаем

$$A_i a_i^z B_i^{w-1} \longrightarrow a_i^{(l-1)\,w+r+1} B_i^{w-1} B_{i0} \ldots B_{i\,w-1} \vdash$$
$$\vdash a_i^{r+1} B_i^{w-1} B_{i0} \ldots B_{i\,w-1} b_i^{(l-1)\,w^2} \vdash$$
$$\vdash B_i^r B_{i0} \ldots B_{i\,w-1} b_i^{lw^2} \vdash B_{i\,w-r} \ldots B_{i\,w-1} b_i^{lw^2+w} \vdash$$
$$\vdash b_i^{lw^2+rw} C_i^w A_\alpha \vdash C_i^w A_\alpha a_\alpha^{cz} \vdash A_\alpha a_\alpha^{cz} B_\alpha^{w-1}.$$

Легко убеждаемся, что полученное соотношение

$$A_i a_i^z B_i^{w-1} \vdash A_\alpha a_\alpha^{cz} B_\alpha^{w-1}$$

верно и для случая $lr = 0$.

2-й случай: i | $:w$ | α | β |.

Пишем продукции

$$A_i \longrightarrow B_{i0} \ldots B_{i\,w-1},\ a_i \longrightarrow b_i b_{i0}^{w-1},\ b_{i0} \longrightarrow a_\alpha^w, \quad (3)$$
$$B_{i0} \longrightarrow C_i^w A_\alpha,\ C_i \longrightarrow B_\alpha^{w-1},\ b_i \longrightarrow a_\alpha,$$
$$B_i \longrightarrow C_i^w,\ B_{i\,w-r} \longrightarrow C_i^{r-1} A_\alpha a_\alpha^r \ (r=1,\ \ldots,\ w-1). \quad (4)$$

Для того чтобы найти, во что эти продукции преобразуют слово $A_i a_i^z B_i^{w-1}$, представим z снова в форме $z = lw + r$ $(0 \leqslant r < w)$. Согласно (3)

$$A_i a_i^{lw+r} B_i^{w-1} \vdash B_i^r B_{i0} \ldots B_{i\,w-1} (b_i b_{i0}^{w-1})^l. \quad (5)$$

Дальнейший ход преобразований зависит от делимости z на w. Пусть z делится на w и, следовательно, $r=0$. В этом случае согласно (4) имеем далее

$$\vdash (b_i b_{i0}^{w-1})^l\, C_i^w A_\alpha \vdash C_i^w A_\alpha a_\alpha^l \vdash A_\alpha a_\alpha^l B_\alpha^{w-1}.$$

Если же $r > 0$, то (5) продолжается следующим образом:

$$\vdash B_{i\,w-r} \ldots B_{i\,w-1} (b_i b_i^{w-1})^l\, C_i^w \vdash$$
$$\vdash (b_{i0}^{w-r} b_i b_{i0}^{r-1})^{l-1}\, b_{i0}^{w-r} C_i^{w+r-1} A_\alpha a_\alpha^r \vdash$$
$$\vdash b_{i0}^{w-r} C_i^{w+r-1} A_\alpha a_\alpha^{r+(l-1)\,w} \vdash C_i^w A_\alpha a_\alpha^z \vdash A_\alpha a_\alpha^z B_\alpha^{w-1}.$$

Итак, если алгоритм (1) пару $(z,\ i)$ преобразует в пару $(u,\ j)$, то продукции (2) — (4) преобразуют слово $A_i a_i^z B_i^{w-1}$ в слово $A_j a_j^u B_j^{w-1}$.

Наконец, для $i = 0$ добавляем продукцию

$$B_0 \xrightarrow{w} a_0 \qquad (6)$$

и называем букву A_0 заключительной.

Обозначим через T однородную систему продукций с заключительной буквой A_0, построенную для программы (1) в соответствии с правилами (2) — (4), (6). Пусть x — произвольное число. Берем слово $A_1 a_1^{2^x} B_1^{w-1}$. Выполняя над ним продукции T, мы будем сначала получать слова, начинающиеся одной из вспомогательных букв B_i, B_{ij}, C_i, b_i, b_{i0}. Первое слово, начинающееся одной из букв A_0, A_1, ..., A_s, будет $A_i a_i^z B_i^{w-1}$, где число z и номер i суть результаты выполнения приказа с номером 1 над числом 2^x. Выполняя над словом $A_i a_i^z B_i^{w-1}$ продукции T, мы снова получим сначала слова, начинающиеся вспомогательными буквами, а за ними получится слово $A_j a_j^u B_j^{w-1}$, где пара (u, j) является результатом выполнения приказа с номером i над числом z и т. д. Если $f(x)$ неопределенное, то процесс будет длиться без конца. Если же $f(x)$ определенное, то после конечного числа продукций мы получим слово $A_0 a_0^{2^{f(x)}} B_0^{w-1}$, отвечающее заключительной паре $(2^{f(x)}, 0)$. Так как буква A_0 заключительная в системе T, то указанное слово будет результатом переработки начального слова посредством системы T. Тем самым первое утверждение теоремы доказано.

Для доказательства второго утверждения мы воспользуемся замечанием, сделанным в п. 14.3 по поводу теоремы 2, согласно которому в формулировке этой теоремы число 5 можно заменить любым числом $p > 1$, не делящимся ни на 2, ни на 3.

Пусть $f(x)$ — частично рекурсивная функция, значения которой не превосходят некоторого числа l. Тогда значения $2^{f(x)}$ будут меньше числа $p = 5^{l+1}$, не делящегося ни на 2, ни на 3. Согласно упомянутому замечанию найдется операторный алгоритм, перерабатывающий 2^x в $2^{f(x)}$, программа которого содержит лишь приказы вида $\boxed{\zeta_i \mid \alpha_i \mid \beta_i}$, где ζ_i принимает значения $\times 2$, $\times 3$, $\times p$, $: 2 \cdot 3 \cdot p$. Выше показано, каким образом можно построить однородную систему продукций T с заключительной буквой A_0, имеющую шаг $w = 6p$ и перерабатывающую слово $A_1 a_1^{2^x} B_1^{w-1}$ в слово $A_0 a_0^{2^{f(x)}} B_1^{w-1}$. Присоединяя к продукциям (2) — (4), (6) системы T еще продукцию

$$A_0 \xrightarrow{w} a_0^{w-1}, \qquad (7)$$

мы обратим T в однородную систему продукций без заключительного элемента, которая слово $A_1 a_1^{2^x} B_1^{w-1}$ преобразует в слово $A_0 a_0^{2^{f(x)}} B_0^{w-1}$. Но теперь последнее слово не является заключитель-

ным и согласно (7), (6) далее имеем

$$A_0 a_0^{2^{f(x)}} B_0^{w-1} \to B_0^{2^{f(x)}} a_0^{w-1} \to a_0^{2^{f(x)}}.$$

Поскольку длина слова $a_0^{2^{f(x)}}$ меньше шага продукций w, то оно теперь и будет результатом переработки начального слова.

С каждой системой продукций T (не обязательно однородной) Э. Пост связал следующие две основные алгоритмические проблемы.

1) Проблема остановки. *Рекурсивна или нет совокупность $\mathfrak{S}$ всех тех слов, начиная с которых процесс T-продукций обрывается через конечное число шагов?*

2) Проблема выводимости. *Для каждого ли слова $\mathfrak{a}$ является рекурсивной совокупность $\mathfrak{S}_\mathfrak{a}$ всех тех слов, которые можно получить из $\mathfrak{a}$ конечным числом T-продукций?*

Э. Пост показал, что существуют такие системы продукций, для которых обе проблемы имеют отрицательное решение (см. п.13.1). Однако вопрос о том, существуют ли *однородные* системы продукций с отрицательно решаемой проблемой остановки, он оставил открытым (ТАГ-проблема). Решение этой более специальной проблемы непосредственно следует из предшествующих результатов.

Теорема 2 (Пост — Минский). *Существует такая система T однородных продукций, для которой совокупность $\mathfrak{S}$ тех слов, начиная с которых процесс T-продукций обрывается, является нерекурсивным множеством.*

Рассмотрим какое-нибудь рекурсивно перечислимое нерекурсивное множество натуральных чисел S и обозначим через $f(x)$ функцию, равную 0 в точках S и не определенную вне S. Функция $f(x)$ частично рекурсивна. Согласно теореме 1 найдется однородная система продукций T, перерабатывающая $A_1 a_1^{2^x} B_1^{w-1}$ в a_0, если $x \in S$, и дающая бесконечный процесс в случае $x \notin S$.

Обозначим через $\mathfrak{M}$ совокупность всех слов вида $A_1 a_1^{2^x} B_1^{w-1}$ и через $\mathfrak{M}_f$ совокупность слов вида $A_1 a_1^{2^u} B_1^{w-1}$, где $u \in S$. Согласно п. 11.1 совокупность $\mathfrak{M}$ рекурсивна, а совокупность $\mathfrak{M}_f$ (вместе с множеством S) нерекурсивна. Но $\mathfrak{M}_f = \mathfrak{S} \cap \mathfrak{M}$, поэтому и совокупность $\mathfrak{S}$ нерекурсивна.

Теорема 3. *Существует такая система однородных продукций T и такое слово $\mathfrak{a}$, что совокупность всех слов $\mathfrak{S}_\mathfrak{a}$, получаемых из $\mathfrak{a}$ конечным числом T-продукций, является нерекурсивной.*

Пусть $g(x)$ — общерекурсивная функция, принимающая различные значения при различных значениях аргумента и имеющая нерекурсивную совокупность S значений. Вводим новую функцию $f(x)$, полагая

$$f(g(n)) = g(n+1) \qquad (n = 0, 1, \ldots).$$

Областью определенности функции $f(x)$ служит множество S, а график ее состоит из пар $\langle g(n), g(n+1) \rangle$ и потому он рекурсивно перечислим. Следовательно, функция $f(x)$ частично рекурсивна. Рассмотрим программу (1) операторного алгоритма, вычисляющего

функцию $f(x)$. Пусть U — последовательность продукций, построенных, исходя из программы (1), с помощью правил (2) — (5). Вместо продукции (6) добавляем к U продукции, получающиеся согласно правилу (2) для приказа

$$0: \boxed{\times 1 \mid 1}\,. \qquad (8)$$

Обозначим через V получившуюся таким путем расширенную систему продукций. Система V однородна. Берем слово $A_1 a_1 B_1^{w-1}$. Из (2) — (5) и (8) получаем

$$A_1 a_1^{2^0} B_1^{w-1} \to \ldots \to A_0 a_0^{2^{f(0)}} B_0^{w-1} \to \ldots \to A_1 a_1^{2^{f(0)}} B_1^{w-1} \to \ldots$$
$$\ldots \to A_0 a^{2^{f(f(0))}} B_0^{w-1} \to \ldots \to A_1 a_1^{2^{f(f(0))}} B_1^{w-1} \vdash \ldots, \qquad (9)$$

причем в последовательности (9) начинаются буквой A_1 лишь явно выписанные слова. Пусть $\mathfrak{a} = A_1 a_1 B_1^{w-1}$, $\mathfrak{S}_\mathfrak{a}$ — совокупность слов, получаемых из $\mathfrak{a}$ конечным числом V-продукций (то есть всех слов последовательности (9)), $\mathfrak{A}$ — совокупность слов вида $A_1 a_1^{2^x} B_1^{w-1}$ и $\mathfrak{B}$ — совокупность слов вида $A_1 a_1^{2^u} B_1^{w-1}$ ($u \in S$). Так как совокупность $\mathfrak{A}$ рекурсивна, $\mathfrak{B}$ нерекурсивна и $\mathfrak{B} = \mathfrak{A} \cap \mathfrak{S}_\mathfrak{a}$, то $\mathfrak{S}_\mathfrak{a}$ нерекурсивна, что и требовалось.

Теоремы 2 и 3 доказаны нами для однородных продукций шага 30. Если воспользоваться теоремой 3 из п. 15.2 и вместо программы (1) рассматривать программы, составленные лишь из приказов вида

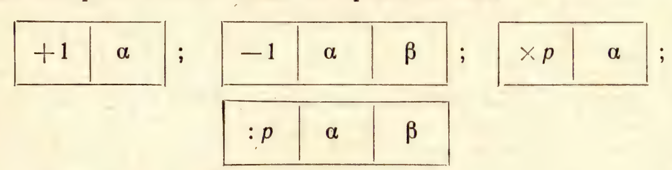
$$\boxed{\times 2 \mid \alpha}\,; \qquad \boxed{\times 3 \mid \alpha}\,; \qquad \boxed{:6 \mid \alpha \mid \beta}\,, \qquad (10)$$

перерабатывающих 2^{2^x} в $2^{2^{f(x)}}$, то продукции, полученные из программ вида (10) согласно правилам (2) — (5), (8), будут иметь шаг, равный 6. Все рассуждения, приведшие нас к теоремам 2 и 3, остаются в силе и для кодирования 2^{2x}, $2^{2f(x)}$. Таким образом, *существуют системы однородных продукций шага, равного* 6, *удовлетворяющие условиям указанных теорем.*

Ясно, что однородных систем продукций шага 1, для которых были бы верны утверждения теорем 2 или 3, нет. Однако, В а н Х а о [12], изменив систему кодирования, построил такую однородную систему продукций, шаг которой равен 2 и для которой проблема остановки не разрешима. Более того, в системе Ван Хао длины всех слов $\mathfrak{b}_1, \ldots, \mathfrak{b}_m$ в преобразованиях T_0 не превышают 3.

Дополнения, примеры и задачи

1. Частичная функция $f(x)$ тогда и только тогда вычислима на двухленточной машине Минского в обычной кодировке, когда $f(x)$ вычислима при помощи операторного алгоритма, программа которого состоит лишь из приказов вида



и стоп-приказа (Я. М. Б а р д з и н ь [5]).

2. В работе [11] Ван Хао рассматривает нестирающие машины Тьюринга, которые могут менять состояния пустых ячеек, но не могут изменять состояний непустых ячеек. Иначе говоря, машина Тьюринга с внешним алфавитом $\{0, 1\}$ называется нестирающей, если она выполняет лишь команды вида

$$q_i 0 \to q_j a T_\varepsilon, \quad q_i 1 \to q_j 1 T_\varepsilon \quad (a = 0,\ 1;\ \varepsilon = 0, \pm 1).$$

Показать, что при подходящем кодировании любая частично рекурсивная функция вычислима на подходящей нестирающей машине Тьюринга (В а н Х а о [11], упрощение и обобщение З ы к и н [37]).

3. Если в однородной системе продукций шага $\mathfrak{w}$ длины всех правых слов $\mathfrak{b}_i$ определяющих продукций не меньше $\mathfrak{w}$ или длина каждого из этих слов не больше $\mathfrak{w}$, то проблемы остановки и выводимости решаются положительно.

4. Пусть однородная система продукций T_0 имеет шаг 1 и при этом некоторые из правых слов $\mathfrak{b}_i$ определяющих продукций могут быть пустыми. Тогда условия предыдущей задачи не выполняются. Тем не менее и в этом случае проблема остановки решается положительно (В а н Х а о [12]).

5. Рассмотрим следующие «упрощенные» приказы. Приказ [$\times 2$] пусть означает — умножить данное число на 2 и перейти к следующему по порядку приказу. Аналогично определим приказ [$\times 3$]. Приказ [$:6$ | α] пусть означает — разделить данное число на 6 и перейти к выполнению приказа с номером α; если же данное число на 6 не делится, то перейти к выполнению над данным числом следующего по порядку приказа. Например, серия упрощенных приказов

$\times 2$	;	$\times 2$	;	$\times 3$	;	$:6$	α

равносильна приказу $\boxed{\times 2 \mid \alpha}$ — умножить данное число на 2 и затем перейти к выполнению приказа с номером α. Аналогично этому серия упрощенных приказов

$$\boxed{:6 \mid \alpha}\ ;\quad \boxed{\times 2}\ ;\quad \boxed{\times 3}\ ;\quad \boxed{:6 \mid \beta}$$

равносильна приказу $\boxed{:6 \mid \alpha \mid \beta}$. С помощью теоремы 3 из п. 15.2. показать, что для каждой частично рекурсивной функции $f(x)$ существует операторный алгоритм, перерабатывающий 2^{2^x} в $2^{2^{f(x)}}$, причем программа этого алгоритма состоит лишь из стоп-приказа и упрощенных приказов вида

$$\boxed{\times 2}\ ,\quad \boxed{\times 3}\quad \text{и}\quad \boxed{:6 \mid \alpha}\ .$$

6. Алгоритм в предыдущей задаче перерабатывает числа. Нетрудно описать соответствующие операторные алгоритмы, перерабатывающие слова в надлежащем алфавите. Например, рассмотрим алфавит, состоящий из цифр 0, 1, 2, 3, 4, 5, 6. Обозначим через $\boxed{i}$ приказ — приписать к данному слову справа цифру i $(0 \leqslant i \leqslant 6)$ и затем перейти к выполнению над полученным словом следующего по порядку приказа.

Через $\boxed{\alpha_0 \mid \alpha_1 \mid \alpha_2 \mid \alpha_3 \mid \alpha_4 \mid \alpha_5 \mid \alpha_6}$ обозначим приказ — в данном слове выбросить первую цифру, перейти далее к выполнению приказа с номером α_i, если отброшенная цифра есть i, и перейти к стоп-приказу, если данное слово пустое.

Например, серия приказов

1 :		6 ,	2	3	4	5	6	7	α
2 :	0 ,	6 ,	2	3	4	5	6	7	α
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
7 :	5 ,	6 ,	2	3	4	5	6	7	α

любое слово $a_0\ a_1\ a_2\ \ldots a_s$ в алфавите 0, 1, 2, 3, 4, 5 перерабатывает в слово $a_0 6 a_1 6 a_2 6 \ldots a_s 6$ и в указание выполнять приказ с номером α. Далее, условимся каждое натуральное число $n \geqslant 1$ кодировать словом $a_0 a_1 \ldots a_s$, где

$$n = a_0 + a_1 \cdot 6 + \ldots + a_s \cdot 6^s \quad (0 \leqslant a_i \leqslant 5;\ a_s \geqslant 1),$$

то есть кодировать «обратной» шестеричной записью n. Тогда приказ

$:6$	α	β

будет равносилен приказу

α	β	β	β	β	β	γ

,

выполненному над «обратной» шестеричной записью данного числа. На основе теоремы 3 из п. 15.2 (см. предыдущую задачу) показать, что для каждой частично рекурсивной функции $f(x)$ существует операторный алгоритм A, перерабатывающий слова в алфавите 0, 1, ..., 6, программа которого состоит из приказов вида

стоп

,

i

,

α_0	α_1	α_2	α_3	α_4	α_5	α_6

и который обратную шестеричную запись произвольного числа вида 2^{2^x} перерабатывает в обратную шестеричную запись числа $2^{2^{f(x)}}$.

7. Рассмотрим слова в двоичном алфавите 0, 1. Приказ

i

($i = 0, 1$) пусть означает — приписать i справа к данному слову и перейти к следующему приказу. Приказ

α_0	α_1

пусть означает — вычеркнуть первую цифру и перейти к приказу с номером α_i, если вычеркнутая цифра есть i ($i = 0, 1$); если данное слово пустое, то остановиться. Беря шестеричную запись натурального числа и заменяя в ней цифры 0, 1, ..., 5 их двоичными разложениями, получим так называемую двоично-шестеричную запись числа. Например, двоично-шестеричная запись числа 2^7 есть 010011011. Показать, что для каждой частично рекурсивной функции $f(x)$ существует алгоритм, перерабатывающий слова в алфавите 0, 1, программа которого состоит лишь из приказов вида

стоп

,

0

,

1

,

α_0	α_1

и который перерабатывает обратную двоично-шестеричную запись числа 2^{2^x} в такую же запись числа $2^{2^{f(x)}}$.

8. Приказ | ╲ | α | пусть означает — вычеркнуть первую букву, перейти к приказу с номером α, если вычеркнут 0, перейти к следующему по порядку приказу, если вычеркнута 1, и остановиться, если заданное слово пустое. Существует операторный алгоритм, перерабатывающий слова в алфавите 0, 1, программа которого состоит лишь из приказов вида | 0 |, | 1 |, | ╲ | α | (без стоп-приказа) и для которого совокупность слов, перерабатываемых в пустое слово, является креативной (Ш е ф е р д с о н — С т у р г и с [118]).

Именно этот результат используется для построения универсальной машины Тьюринга с символами a_0, a_1, a_2, a_3 и внутренними состояниями q_0, q_1, . . . , q_6, пока «наименьшей» из известных универсальных машин Тьюринга (Т р и т т е р [100]).

9. Каково минимальное значение n, для которого существует универсальная машина Тьюринга $\mathfrak{T}$ с символами 0, 1 и состояниями q_0, q_1, . . . , q_n? Каковы минимальные n, для которых существует такая машина $\mathfrak{T}$, что $\mathfrak{T}_{\mathrm{fin}}$ нерекурсивное, креативное, простое, максимальное?

10. Согласно Б ю х и [10], *регулярной* системой над алфавитом $\{c_1, \ldots, c_m\}$ называется система, определяемая указанным алфавитом и системой базисных преобразований вида

$$\mathfrak{a}_i W \to \mathfrak{b}_i W \quad (i=1, \ldots, s), \tag{A}$$

где $\mathfrak{a}_i$, $\mathfrak{b}_i$ — фиксированные слова в упомянутом алфавите.

Бюхи показал, что совокупность слов M тогда и только тогда порождается конечной системой слов при помощи преобразований вида (A), когда M порождается конечной системой слов при помощи преобразований вида

$$\mathfrak{a}_i W \to \mathfrak{a}_i \mathfrak{p}_i W \quad (\mathfrak{p}_i \neq \Lambda).$$

Отсюда, в частности, следует, что все совокупности, порождаемые в *регулярных* системах конечными системами слов, являются рекурсивными.

§ 16. ДИОФАНТОВЫ УРАВНЕНИЯ

В 1900 г. на Международном математическом конгрессе Д. Г и л ь б е р т [16] выделил 23 проблемы, решение которых представило бы особый интерес для развития математики.

Среди этих проблем помещена и проблема № 10, сформулированная Гильбертом следующим образом:

10. Выяснение разрешимости произвольного диофантова уравнения. *Пусть задано диофантово уравнение с произвольным числом неизвестных и целыми рациональными коэффициентами; требуется указать способ, по которому с помощью конечного числа операций можно было бы узнать, разрешимо уравнение в целых рациональных числах или нет.*

Способ (Verfahren), об отыскании которого идет речь в этой проблеме, мы понимаем теперь как алгоритм. В 1900 г. теория алгоритмов еще не была создана и речь могла идти лишь о положительном решении проблемы. Однако большие трудности, возникающие обычно при исследовании диофантовых уравнений, привели к предположению, что упомянутый алгоритм не существует. Ниже будут изложены результаты М. Девиса, Х. Путнама и Ю. Робинсон [26], показавших, что ряд арифметических проблем, близких по своему характеру 10-й проблеме Гильберта, решается отрицательно. Что касается самой проблемы Гильберта, то она пока еще остается открытой.

16.1. Диофантовы предикаты и функции. В 10-й проблеме Гильберта речь идет о существовании целых рациональных решений, то есть решений в числах 0, ± 1, ± 2, ... Покажем, что эта проблема равносильна проблеме существования решения в натуральных числах 0, 1, 2, ... В самом деле, пусть для *каждого* уравнения

$$F(x_1, \ldots, x_n) = 0, \tag{1}$$

где F — многочлен с целыми рациональными коэффициентами, мы умеем решить вопрос, имеет оно или нет решение в *натуральных* числах. Спрашивается, имеет ли уравнение (1) решение в целых рациональных числах? Составляем уравнение

$$F(x_1, \ldots, x_n) \cdot F(-x_1, \ldots, x_n) \times \ldots \\ \ldots \times F(-x_1, \ldots, -x_n) = 0. \tag{2}$$

Ясно, что уравнение (1) тогда и только тогда разрешимо

в целых рациональных числах, когда уравнение (2) разрешимо в натуральных числах.

Обратно, пусть мы умеем решать вопрос, имеет или нет решение в целых рациональных числах произвольно заданное уравнение вида (1). Спрашивается, имеет ли уравнение (1) решение в натуральных числах? Согласно общеизвестной теореме Лагранжа, каждое натуральное число представимо в виде суммы квадратов четырех натуральных чисел. Поэтому уравнение (1) тогда и только тогда имеет решение в натуральных числах, когда уравнение

$$F(x_1^2+y_1^2+z_1^2+u_1^2, \ldots, x_n^2+y_n^2+z_n^2+u_n^2)=0$$

имеет решение $\langle x_1, y_1, z_1, u_1, \ldots, x_n, y_n, z_n, u_n \rangle$ в целых рациональных числах. Ввиду изложенного, ниже будет рассматриваться лишь задача о существовании решений в натуральных числах и символы $(\exists x)$, $(\forall x)$, $(\forall x \leqslant a)$ будут означать соответственно «существует натуральное число x», «для каждого натурального числа x», «для каждого натурального x из отрезка 0, 1, 2, ..., a».

Предикат $P(x_1, \ldots, x_n)$, определенный на множестве натуральных чисел, называется *диофантовым*, если существует такой многочлен $F(x_1, \ldots, x_n, y_1, \ldots, y_m)$ с целыми коэффициентами, что

$$P(x_1 \ldots, x_n) \Leftrightarrow$$
$$\Leftrightarrow (\exists y_1) \ldots (\exists y_m)(F(x_1, \ldots, x_n, y_1, \ldots, y_m)=0). \qquad (3)$$

Множество $\boldsymbol{M}$ n-ок $\langle x_1, \ldots, x_n \rangle$ натуральных чисел называется *диофантовым*, если *диофантовым* является n-местный предикат P, истинный на n-ках из $\boldsymbol{M}$ и ложный на остальных n-ках натуральных чисел.

Из соотношения (3) видно, что каждое диофантово множество рекурсивно перечислимо. Если бы проблема Гильберта решалась положительно, то каждое диофантово множество было бы рекурсивным. Поэтому для отрицательного решения проблемы Гильберта достаточно показать, что существуют нерекурсивные диофантовы множества. Однако пока неизвестно, существуют ли нерекурсивные диофантовы множества.

Для краткости условимся символами $\mathfrak{x}^{(n)}$, $\mathfrak{y}^{(n)}$ обозначать соответственно n-ки $\langle x_1, \ldots, x_n \rangle$, $\langle y_1, \ldots, y_n \rangle$.

Из только что введенных определений непосредственно вытекает

Следствие 1. *Если предикаты* $P(\mathfrak{x}^{(n)})$, $Q(\mathfrak{x}^{(n)})$, $R(\mathfrak{x}^{(n)}, \mathfrak{y}^{(m)})$ *диофантовы, то предикаты* $P(\mathfrak{x}^{(n)})\ \&\ Q(\mathfrak{x}^{(n)})$, $P(\mathfrak{x}^{(n)}) \vee Q(\mathfrak{x}^{(n)})$, $(\exists y_1 \ldots y_m)\, R(\mathfrak{x}^{(n)}, \mathfrak{y}^{(m)})$ *также диофантовы. Если* $F(\mathfrak{x}^{(n)})$, $G(\mathfrak{x}^{(n)})$ — *многочлены с целыми коэффициентами, то предикаты* $S(\mathfrak{x}^{(n)})$, $T(\mathfrak{x}^{(n)})$ $U(\mathfrak{x}^{(n)})$, *определенные соотношениями*

$$S(\mathfrak{x}^{(n)}) \iff F(\mathfrak{x}^{(n)}) = G(\mathfrak{x}^{(n)}),$$

$$T(\mathfrak{x}^{(n)}) \iff F(\mathfrak{x}^{(n)}) \neq G(\mathfrak{x}^{(n)}),$$

$$U(\mathfrak{x}^{(n)}) \iff F(\mathfrak{x}^{(n)}) < G(\mathfrak{x}^{(n)}),$$

диофантовы.

В самом деле, пусть

$$P(\mathfrak{x}^{(n)}) \iff (\exists \mathfrak{y}^{(m)})\,(H(\mathfrak{x}^{(n)}, \mathfrak{y}^{(m)}) = 0),$$

$$Q(\mathfrak{x}^{(n)}) \iff (\exists \mathfrak{z}^{(k)})\,(L(\mathfrak{x}^{(n)}, \mathfrak{z}^{(k)}) = 0),$$

где H, L — многочлены с целыми коэффициентами. Тогда

$$P(\mathfrak{x}^{(n)})\ \&\ Q(\mathfrak{x}^{(n)}) \iff (\exists \mathfrak{y}^{(m)})\,(\exists \mathfrak{z}^{(k)})\,(H + L = 0),$$

$$P(\mathfrak{x}^{(n)}) \vee Q(\mathfrak{x}^{(n)}) \iff (\exists \mathfrak{y}^{(m)})\,(\exists \mathfrak{z}^{(k)})\,(H \cdot L = 0).$$

Далее, имеем

$$T(\mathfrak{x}^{(n)}) \iff (\exists y)\,((F - G)^2 = y + 1),$$

$$U(\mathfrak{x}^{(n)}) \iff (\exists y)\,(1 + y + F = G),$$

что и требовалось.

Числовая функция $f(x_1, \ldots, x_n)$ называется диофантовой, если $(n+1)$-местное отношение $y = f(x_1, \ldots, x_n)$ диофантово.

В частности, отсюда вытекает, что каждый многочлен с целыми коэффициентами является диофантовой функцией от своих аргументов.

Теорема 1. *Суперпозиция диофантовых функций есть функция диофантова. Если предикат* $P(\mathfrak{x}^{(n)})$ *и функции* $F_1(\mathfrak{x}^{(m)}), \ldots, F_n(\mathfrak{x}^{(m)})$ *диофантовы, то предикат* $P(F_1(\mathfrak{x}^{(m)}), \ldots, F_n(\mathfrak{x}^{(m)}))$ *также диофантов.*

Действительно, пусть функция $F(\mathfrak{x}^{(n)})$ и функции $F_i(\mathfrak{x}^{(m)})$ диофантовы. Так как отношение

$$y = F(F_1(\mathfrak{x}^{(m)}), \ldots, F_n(\mathfrak{x}^{(m)}))$$

для натуральных $x_1, \ldots, x_m$ равносильно отношению

$$(\exists z_1 \ldots z_n) \prod_{\varepsilon_1, \ldots, \varepsilon_n = \pm 1} ((y - F(\varepsilon_1 z_1, \ldots, \varepsilon_n z_n))^2 + \\ + \sum (\varepsilon_i z_i - F_i(\mathfrak{x}^{(m)}))^2) = 0,$$

а это последнее отношение в силу следствия 1 диофантово, то функция $F(F_1, \ldots, F_n)$ диофантова. Аналогично доказывается и второе утверждение теоремы.

Т е о р е м а 2. *Функции* $x \dot{-} y$, $[x/y]$, $\mathrm{rest}(x, y)$, $\boldsymbol{c}(x, y)$, $\boldsymbol{l}(x)$, $\boldsymbol{r}(x)$, $\boldsymbol{\Gamma}(x, y)$ *диофантовы.*

В самом деле,

$$z = x \dot{-} y \Leftrightarrow (x \leqslant y \,\&\, z = 0) \vee x = z + y,$$

$$z = [x/y] \Leftrightarrow (y = 0 \,\&\, z = x) \vee (\exists u)(x = yz + u \,\&\, u < y),$$

$$\mathrm{rest}(x, y) = x \dot{-} y \cdot [x/y], \quad \boldsymbol{c}(x, y) = \left[\frac{(x+y)^2 + 3x + y}{2}\right],$$

$$z = \boldsymbol{l}(x) \Leftrightarrow (\exists v)(\boldsymbol{c}(z, v) = x),$$

$$z = \boldsymbol{r}(x) \Leftrightarrow (\exists v)(\boldsymbol{c}(v, z) = x),$$

$$\boldsymbol{\Gamma}(x, y) = \mathrm{rest}(\boldsymbol{l}(x), 1 + (y+1)\,\boldsymbol{r}(x)).$$

Применяя следствие 1 и теорему 1, непосредственно убеждаемся, что стоящие справа от знака $\Leftrightarrow$ формулы представляют диофантовы предикаты.

Обозначим через $\boldsymbol{A}$ совокупность всех степеней числа 2 : $\boldsymbol{A} = \{1, 2, 4, 8, \ldots\}$ и пусть $\boldsymbol{A}' = \boldsymbol{N} - \boldsymbol{A}$. Интересно отметить, что множество $\boldsymbol{A}'$ диофантово, так как

$$x \in \boldsymbol{A}' \Leftrightarrow (\exists uv)(x = (2u + 3)\,v).$$

В то же время пока неизвестно, является ли диофантовым множество $\boldsymbol{A}$. В частности, неизвестно, является ли диофантовой функция 2^x (ср. Ю. Р о б и н с о н [88]). В настоящее время, по-видимому, неизвестен ни один конкретный диофантов предикат, имеющий недиофантово отрицание, хотя очень просто доказывается (неконструктивно)

Т е о р е м а 3. *Существуют диофантовы предикаты, отрицание которых недиофантово*

Допустим противное, то есть допустим, что для каждого многочлена $F(\mathfrak{x}, \mathfrak{y})$ с целыми коэффициентами существует

такой многочлен $F^*(\mathfrak{x}, \mathfrak{y}_1)$ с целыми коэффициентами, что

$$\neg (\exists \mathfrak{y}) (F (\mathfrak{x}, \mathfrak{y}) = 0) \Leftrightarrow (\exists \mathfrak{y}_1) (F^* (\mathfrak{x}, \mathfrak{y}_1) = 0) \qquad (4)$$

и, следовательно,

$$(\exists \mathfrak{y}) (F (\mathfrak{x}, \mathfrak{y}) = 0) \Leftrightarrow (\forall \mathfrak{y}_1) (F^* (\mathfrak{x}, \mathfrak{y}_1) \neq 0). \qquad (5)$$

Покажем, что в таком случае каждое арифметическое множество (п. 13.3) будет диофантовым. Рассмотрим, например, предикат

$$(\forall \mathfrak{y}) (\exists \mathfrak{z}) (G (\mathfrak{x}, \mathfrak{y}, \mathfrak{z}) = 0),$$

где G — многочлен. На основании (4), (5) имеем

$$(\forall \mathfrak{y}) (\exists \mathfrak{z}) (G (\mathfrak{x}, \mathfrak{y}, \mathfrak{z}) = 0) \Leftrightarrow (\forall \mathfrak{y}) (\forall \mathfrak{z}_1) (G^* (\mathfrak{x}, \mathfrak{y}, \mathfrak{z}_1) \neq 0) \Leftrightarrow$$
$$\Leftrightarrow \Box (\exists \mathfrak{y}) (\exists \mathfrak{z}_1) (G^* (\mathfrak{x}, \mathfrak{y}, \mathfrak{z}_1) = 0) \Leftrightarrow (\exists \mathfrak{z}_2) (G^{**} (\mathfrak{x}, \mathfrak{z}_2) = 0).$$

Изложенные преобразования показывают, что и в формулах с более сложными приставками вида $\forall\exists$, $\exists\forall\exists$ и т. д. кванторы всеобщности можно постепенно заменить кванторами существования и тем самым каждый арифметический предикат представить в диофантовом виде. Мы пришли к противоречию, так как все диофантовы предикаты рекурсивно перечислимы, а среди арифметических предикатов согласно п. 13.3 есть и не рекурсивно перечислимые.

Существует предположение, что все рекурсивно перечислимые предикаты диофантовы. Указанные в теореме 2 функции имеют медленный рост. Ю. Робинсон [88] показала, что если существует хотя бы одна диофантова функция, растущая не медленнее 2^x, то все рекурсивно перечислимые предикаты диофантовы и проблема Гильберта решается отрицательно.

В связи с теоремой 3 введем следующее понятие, которое будет нам полезно ниже.

Предикат $P(x_1, \ldots, x_m)$ условимся называть *дважды диофантовым*, если он сам и его отрицание диофантовы. Соответственно этому, функцию $f(x_1, \ldots, x_n)$ будем называть *дважды диофантовой*, если отношение $f(x_1, \ldots$ $\ldots, x_n) = x_{n+1}$ дважды диофантово.

Из этого определения видно, что предикат $P(x_1, \ldots, x_m)$ тогда и только тогда дважды диофантов, когда существуют

такие многочлены F, G с целыми коэффициентами, что

$$P(x_1, \ldots, x_m) \Leftrightarrow$$
$$\Leftrightarrow (\exists y_1, \ldots, y_s)(F(x_1, \ldots, x_m; y_1, \ldots, y_s) = 0), \quad (6)$$
$$P(x_1, \ldots, x_m) \Leftrightarrow$$
$$\Leftrightarrow (\forall z_1) \ldots (\forall z_t)(G(x_1, \ldots, x_m, z_1, \ldots, z_t) \neq 0). \quad (7)$$

Так как дважды диофантово множество и его дополнение диофантовы, а диофантовы множества рекурсивно перечислимы, то каждое дважды диофантово множество рекурсивно. Верно ли обратное — неизвестно. Теорема 3 утверждает существование диофантова множества, не являющегося дважды диофантовым. Если бы все рекурсивные множества были дважды диофантовы, то упомянутое диофантово множество с недиофантовым дополнением было бы нерекурсивным и проблема Гильберта решалась бы отрицательно.

Изложим несколько простых свойств дважды диофантовых множеств и функций.

Теорема 4. *Конъюнкция, дизъюнкция и отрицание дважды диофантовых предикатов являются дважды диофантовыми предикатами. Суперпозиция дважды диофантовых всюду определенных функций есть дважды диофантова функция. Если предикат $P(x_1, \ldots, x_m)$ и функция $f(x_1, \ldots, x_m)$ дважды диофантовы, то предикат $P(f(x_1, \ldots, x_m), x_2, \ldots, x_m)$ также дважды диофантов.*

Докажем, например, последнее утверждение. Имеем

$$P(f(x_1, \ldots, x_m), \ldots, x_m) \Leftrightarrow$$
$$\Leftrightarrow (\exists u)(f(x_1, \ldots, x_m) = u \,\&\, P(u, x_2, \ldots, x_m)), \quad (8)$$
$$P(f(x_1, \ldots, x_m), \ldots, x_m) \Leftrightarrow$$
$$\Leftrightarrow (\forall u)(f(x_1, \ldots, x_m) \neq u \vee P(u, x_2, \ldots, x_m)). \quad (9)$$

По условию существуют такие многочлены F, G с целыми коэффициентами, что

$$f(x_1, \ldots, x_m) = u \,\&\, P(u, x_2, \ldots, x_m) \Leftrightarrow$$
$$\Leftrightarrow (\exists y_1 \ldots y_s)(F(u, x_1, \ldots, x_m; y_1, \ldots, y_s) = 0),$$
$$f(x_1, \ldots, x_m) \neq u \vee P(u, x_2, \ldots, x_m) \Leftrightarrow$$
$$\Leftrightarrow (\forall z_1) \ldots (\forall z_t)(G(u, x_1, \ldots, x_m, z_1, \ldots, z_t) \neq 0).$$

Сравнивая эти эквивалентности с соответственными эквивалентностями (8), (9), видим, что предикат $P\ (f\ (x_1, \ldots, x_m), x_2, \ldots, x_m)$ представим в формах (6), (7) и, следовательно, этот предикат дважды диофантов.

Теорема 5. *Каждый многочлен с целыми коэффициентами является дважды диофантовой функцией. Отношения* $x < y$, $x = y$ *и функции* $[x/y]$, $\operatorname{rest}(x, y)$, $\boldsymbol{l}(x)$, $\boldsymbol{r}(x)$, $\Gamma(x, y)$ *дважды диофантовы.*

Действительно,

$$y = f(\mathfrak{x}) \Leftrightarrow y - f(\mathfrak{x}) = 0 \Leftrightarrow (\forall z)(z + 1 - (y - f(\mathfrak{x}))^2 \neq 0),$$

$$x < y \Leftrightarrow (\exists z)(x + z + 1 - y = 0) \Leftrightarrow (\forall z)(x - y - z \neq 0),$$

$$z = [x/y] \Leftrightarrow (zy \leqslant x \mathbin{\&} x < (z+1)y) \vee (y = 0 \mathbin{\&} z = x),$$

$$\operatorname{rest}(x, y) = x - [x/y]\, y,$$

$$y = \boldsymbol{l}(x) \Leftrightarrow (\exists z)(c(y, z) - x = 0) \Leftrightarrow$$
$$\Leftrightarrow (\forall z)(\forall u)(c(z, u) \neq x \vee z = y),$$

$$\Gamma(x, y) = \operatorname{rest}(\boldsymbol{l}(x), 1 + (y+1)\, \boldsymbol{r}(x)).$$

Ясно, что если предикат $P(x_1, \ldots, x_m)$ дважды диофантов, то предикат $(\exists x_1)\, P(x_1, \ldots, x_m)$ диофантов. Однако ввиду теоремы 3 последний предикат, вообще говоря, не будет дважды диофантовым.

16.2. Арифметическое представление. Уже говорилось, что существует гипотеза, согласно которой каждое рекурсивно перечислимое множество допускает диофантово представление. Вопрос об истинности этой гипотезы пока открыт. Однако можно доказать, что каждое рекурсивно перечислимое множество допускает арифметическое представление, отличающееся от диофантова лишь двумя кванторами. Это представление лежит в основе вывода ряда других представлений для рекурсивно перечислимых множеств.

Теорема 1 (Девис [23]). *Для каждого рекурсивно перечислимого множества* $\boldsymbol{M}$ *существует такой многочлен* $F(x, y, z, x_1, \ldots, x_n)$ *с целыми коэффициентами, что*

$$x \in \boldsymbol{M} \Leftrightarrow (\exists y)(\forall z \leqslant y)(\exists x_1) \ldots$$
$$\ldots (\exists x_n)(F(x, y, z, x_1, \ldots, x_n) = 0), \quad (1)$$

где кванторы $\exists$, $\forall$, как и всюду, относятся к множеству натуральных чисел.

Посмотрим сначала, что дает эта теорема для проблемы решения алгебраических уравнений в натуральных числах. Берем в качестве $\boldsymbol{M}$ какое-нибудь нерекурсивное рекурсивно перечислимое множество и строим соответствующий многочлен F. Рассмотрим следующую задачу: для произвольно заданного натурального x узнать, будет ли разрешима в натуральных y, x_{ij} система уравнений *)

$$\left.\begin{array}{l} F(x, y, 0, x_{01}, \ldots, x_{0n}) = 0, \\ F(x, y, 1, x_{11}, \ldots, x_{1n}) = 0, \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ F(x, y, y, x_{y1}, \ldots, x_{yn}) = 0. \end{array}\right\} \qquad (2)$$

Согласно теореме 1 разрешимость этой системы равносильна принадлежности числа x множеству $\boldsymbol{M}$. Значит, задача о разрешимости указанной системы для произвольно заданного числа x не имеет алгоритма для своего решения.

Перейдем к доказательству теоремы 1. Рассмотрим функцию $f(x)$, равную 0 для $x \in \boldsymbol{M}$ и неопределенную для $x \notin \boldsymbol{M}$. Эта функция частично рекурсивная и потому согласно теореме 2 из п. 14.3 существует операторный алгоритм, перерабатывающий 2^x в $2^{f(x)}$ и имеющий программу вида

$$0\colon\ \boxed{\text{стоп}}\ ;\quad 1\colon\ \boxed{\zeta_1 \mid a_1 \mid b_1}\ ;\ \ldots;\quad n\colon\ \boxed{\zeta_n \mid a_n \mid b_n}\ , \qquad (3)$$

где ζ_i означает одну из операций $\times 2$, $\times 3$, $\times 5$, $:30$. Эти операции перерабатывают число вида $2^u \cdot 3^v \cdot 5^w$ в число того же вида. Поэтому, начиная согласно указанной программе перерабатывать пару $(2^x, 1)$, мы после n-го шага вычислений будем иметь пару вида $(2^{u_s} \cdot 3^{v_s} \cdot 5^{w_s},\ c_s)$, где $2^{u_s}3^{v_s}5^{w_s}$ — полученное число, а c_s — номер приказа в программе (3), который следует выполнить на следующем шаге.

Обозначим через $P_1(u, v, w, c, u', v', w', c')$ предикат, истинный тогда и только тогда, когда пара $(2^u \cdot 3^v \cdot 5^w, c)$

*) Число уравнений в этой системе, равное $y+1$, тоже является искомым.

переходит в пару $(2^{u'}\cdot 3^{v'}\cdot 5^{w'}, c')$ в результате выполнения приказа с номером c из программы (3). При этом будем считать $P_1(u, v, w, c, u', v', w', c')$ ложным для любых u', v', w', c', если $c > n$; а для $c = 0$ положим P_1 истинным лишь тогда, когда $c' = 0$ и $u = u'$, $v = v'$, $w = w'$.

Покажем, что предикат P_1 диофантов. Для каждого i, $1 \leqslant i \leqslant n$, введем предикат $P_1^i(u, v, w, c; u', v', w', c')$, равносильный утверждению

$$c = i \mathbin{\&} P_1(u, v, w, c; u', v', w', c'),$$

и покажем, что предикаты $P_1^0, P_1^1, \ldots, P_1^n$ диофантовы. Для P_1^0 это очевидно, так как согласно сказанному

$$P_1^0(u, v, w, c; u', v', w', c') \Leftrightarrow$$
$$\Leftrightarrow c = c' = 0 \mathbin{\&} u = u' \mathbin{\&} v = v' \mathbin{\&} w = w'.$$

Рассмотрим P_1^i для $i \geqslant 1$. В соответствующем приказе i: | ζ_i | a_i | b_i | символ ζ_i может иметь одно из четырех значений: $\times 2$, $\times 3$, $\times 5$, $:30$. Допустим, что упомянутый приказ имеет вид

i: | :30 | a_i | b_i | .

Следовательно, если $c = i$ и число $2^u\cdot 3^v\cdot 5^w$ делится на 30, то пара $(2^u\cdot 3^v\cdot 5^w, c)$ перейдет в пару $(2^{u-1}\cdot 3^{v-1}\cdot 5^{w-1}, a_i)$. Если же указанное число не делится на 30, то упомянутая пара перейдет в пару $(2^u\cdot 3^v\cdot 5^w, b_i)$. Иначе говоря, в рассматриваемом случае отношение $P_1^i(u, v, w, c; u', v', w', c')$ равносильно выражению

$$c = i \mathbin{\&} (u > 0 \mathbin{\&} v > 0 \mathbin{\&} w > 0 \to u' = u - 1 \mathbin{\&} v' =$$
$$= v - 1 \mathbin{\&} w' = w - 1 \mathbin{\&} c' = a_i) \mathbin{\&} (u\cdot v\cdot w = 0 \to u' =$$
$$= u \mathbin{\&} v' = v \mathbin{\&} w' = w \mathbin{\&} c' = b_i),$$

эквивалентному формуле

$$c = i \mathbin{\&} u\cdot v\cdot w = 0 \vee (u = u' + 1 \mathbin{\&} v = v' + 1 \mathbin{\&} w =$$
$$= w' + 1 \mathbin{\&} c' = a_i) \mathbin{\&} (u > 0 \mathbin{\&} v > 0 \mathbin{\&} w > 0) \vee (u =$$
$$= u' \mathbin{\&} v = v' \mathbin{\&} w = w' \mathbin{\&} c' = b_i).$$

Согласно предшествующему пункту предикаты $c = i$, $u > 0$, $u = u' + 1$ и т. п. диофантовы, конъюнкции и дизъ-

юнкции их также диофантовы. Следовательно, предикат P_i^i диофантов.

Допустим теперь, что рассматриваемый приказ имеет вид

$$i: \boxed{\times 2 \mid a_i}$$

В этом случае пара $(2^u \cdot 3^v \cdot 5^w,\ c)$ переходит в пару $(2^{u+1} \cdot 3^v \cdot 5^w,\ a_i)$, предикат $P_1^i(u,\ v,\ w,\ c;\ u',\ v',\ w',\ c')$ записывается в виде

$$c = i \,\&\, u' = u + 1 \,\&\, v' = v \,\&\, w' = w \,\&\, c' = a_i$$

и потому является диофантовым.

Случаи $\zeta_i = \times 3,\ \times 5$ рассматриваются аналогично. Поэтому можно считать, что диофантовость всех предикатов P_1^i доказана. Но

$$P_1 \Leftrightarrow P_1^0 \vee P_1^1 \vee \ldots \vee P_1^n,$$

поэтому предикат P_1 также диофантов.

Обозначим через $P_s(u,\ v,\ w,\ c;\ u',\ v',\ w',\ c')$ предикат, который при $c \leqslant n$ истинен тогда и только тогда, когда пара $(2^u \cdot 3^v \cdot 5^w,\ c)$ переходит в пару $(2^{u'} \cdot 3^{v'} \cdot 5^{w'},\ c')$ после s шагов переработки по программе (3). Вводя обозначения $\mathfrak{x}_i = \langle x_{i1},\ x_{i2},\ x_{i3},\ x_{i4} \rangle$ и принимая во внимание смысл отношения P_1, видим, что отношение $P_{s+1}(\mathfrak{x},\ \mathfrak{y})$ равносильно формуле

$$(\exists \mathfrak{x}_1 \mathfrak{x}_2 \ldots \mathfrak{x}_s)(P_1(\mathfrak{x},\ \mathfrak{x}_1) \,\&\, P_1(\mathfrak{x}_1, \mathfrak{x}_2) \,\&\, \ldots \,\&\, P_1(\mathfrak{x}_s,\ \mathfrak{y})). \quad (4)$$

Согласно основному свойству функции Геделя $\mathbf{\Gamma}(z,\ x)$ (п. 3.3) для любых натуральных x_{ij} $(i = 0, 1, \ldots, s+1;\ j = 1,\ 2,\ 3,\ 4)$ существует такое a, что

$$\mathbf{\Gamma}(a,\ i) = \boldsymbol{c}^4(x_{i1},\ x_{i2},\ x_{i3},\ x_{i4}) \qquad (i = 0,\ 1,\ \ldots,\ s+1),$$

и, следовательно,

$$x_{ij} = \boldsymbol{c}_{4j}(\mathbf{\Gamma}(a,\ i)) \quad (i = 0,\ 1,\ \ldots,\ s+1;\ j = 1,\ 2,\ 3,\ 4), \quad (5)$$

где $\boldsymbol{c}^4$, $\boldsymbol{c}_{41}$, $\boldsymbol{c}_{42}$, $\boldsymbol{c}_{43}$, $\boldsymbol{c}_{44}$ — нумерационные функции, определенные в п. 3.3, причем $\mathfrak{x}_0 = \mathfrak{x}$, $\mathfrak{x}_{s+1} = \mathfrak{y}$. Все эти функции являются суперпозициями функций $\boldsymbol{c}$, $\boldsymbol{l}$, $\boldsymbol{r}$, диофантовость которых была установлена в предшествующем пункте. Отсюда вытекает, что функции $\boldsymbol{c}_{4j}(\mathbf{\Gamma}(x,\ y))$ также диофантовы.

Из разрешимости относительно a системы (5) следует, что формула (4) равносильна утверждению

$$(\exists a)(\forall i \leqslant s)(P_1(d_1(a, i), \ldots, d_4(a, i), d_1(a, i+1), \ldots \\ \ldots, d_4(a, i+1)) \,\&\, x_1 = d_1(a, 0) \,\&\, \ldots \,\&\, x_4 = d_4(a, 0) \,\&\, y_1 = \\ = d_1(a, s+1) \,\&\, \ldots \,\&\, y_4 = d_4(a, s+1)), \quad (6)$$

где $d_j(a, i) = c_{4j}(\mathbf{\Gamma}(a, i))$.

Условие $x \in \boldsymbol{M}$ означает, что процесс переработки четверки $(x, 0, 0, 1)$ на подходящем $(s+1)$-м шаге обрывается, то есть найдется такое s, что в получающейся на $(s+1)$-м шаге четверке (y_1, y_2, y_3, y_4) имеем $y_4 = 0$. Иначе говоря, формула (6) дает

$$x \in \boldsymbol{M} \Leftrightarrow (\exists s)(\exists a)(\forall i \leqslant s)\{P_1(d_1(a, i), \ldots \\ \ldots, d_4(a, i), d_1(a, i+1), \ldots, d_4(a, i+1)) \,\&\, x = d_1(a, 0) \,\&\, \\ \&\, 0 = d_2(a, 0) \,\&\, 0 = d_3(a, 0) \,\&\, 1 = d_4(a, 0) \,\&\, 0 = d_4(a, s+1)\}. \quad (7)$$

Мы уже видели, что предикат P_1, функции $d_2(a, i)$ и отношения $x = d_1(a, 0), \ldots, 0 = d_4(a, s+1)$ диофантовы. Поэтому выражение, стоящее в формуле (7) внутри фигурных скобок, представляет диофантов предикат от переменных x, i, a, s. Обозначая этот предикат через Q, имеем

$$x \in \boldsymbol{M} \Leftrightarrow (\exists s)(\exists a)(\forall i \leqslant s)\, Q(x, a, i, s). \quad (8)$$

Но утверждение о существовании пары чисел s, a равносильно утверждению о существовании числа $y = \boldsymbol{c}(s, a)$, являющегося номером этой пары. Поэтому из соотношения (8) получаем

$$x \in \boldsymbol{M} \Leftrightarrow (\exists y)(\forall i \leqslant \boldsymbol{l}(y))\, Q(x, \boldsymbol{r}(y), i, \boldsymbol{l}(y)) \Leftrightarrow \\ \Leftrightarrow (\exists y)(\forall i \leqslant y)(Q(x, \boldsymbol{r}(y), i, \boldsymbol{l}(y)) \vee i > \boldsymbol{l}(y)). \quad (9)$$

Предикат $Q(x, \boldsymbol{r}(y), i, \boldsymbol{l}(y)) \vee i > \boldsymbol{l}(y)$ диофантов и, следовательно, его можно представить в виде

$$(\exists x_1, \ldots, x_n)(F(x, y, i, x_1, \ldots, x_n) = 0), \quad (10)$$

где F — подходящий многочлен с целыми коэффициентами. Соединяя формулы (9) и (10), приходим к требуемой формуле (1).

С помощью небольших формальных преобразований (Девис [23]) из теоремы 1 легко получается.

Теорема 2. *Для каждого рекурсивно перечислимого множества* $\boldsymbol{M}$ *существует такой многочлен* G *с целочисленными коэффициентами, что*

$$\in \boldsymbol{M} \Leftrightarrow (\exists y)(\forall z \leqslant y)(\exists x_1 \leqslant y)\ldots$$
$$\ldots(\exists x_m \leqslant y)(G(x, y, z, x_1, \ldots, x_m) = 0).$$

Согласно теореме 1 существует многочлен F, удовлетворяющий соотношению (1), и потому утверждение $x \in \boldsymbol{M}$ равносильно разрешимости при некотором y системы уравнений (2) относительно x_{ij} ($i = 0, \ldots, y$; $j = 1, \ldots, n$). Обозначая через u число, большее каждого числа x_{ij}, видим, то соотношение (1) влечет

$$x \in \boldsymbol{M} \Leftrightarrow (\exists y)(\exists u)(\forall z \leqslant y)\ (\exists x_1 \leqslant u)\ldots(\exists x_n \leqslant u)(F = 0).$$

Вводя вместо пары чисел y, u ее номер $v = \boldsymbol{c}(y, u)$, приходим к заключению, что отношение $x \in \boldsymbol{M}$ равносильно формуле

$$(\exists v)(\forall z \leqslant \boldsymbol{l}(v))(\exists x_1 \leqslant \boldsymbol{r}(v))\ldots$$
$$\ldots(\exists x_n \leqslant \boldsymbol{r}(v))(F(x, \boldsymbol{l}(v), z, x_1, \ldots, x_n) = 0),$$

которая, в свою очередь, равносильна выражению

$$(\exists v)(\forall z \leqslant v)(\exists x_1 \leqslant v)\ldots(\exists x_n \leqslant v)(\exists w_1 \leqslant v)$$
$$(\exists w_2 \leqslant v)(\boldsymbol{c}(w_1, w_2) = v \,\&\, [(F(x, w_1, z, x_1, \ldots, x_n) = 0) \vee$$
$$\vee (w_1 < z \leqslant v)] \,\&\, x_1 \leqslant w_2 \,\&\, \ldots \,\&\, x_n \leqslant w_2). \quad (11)$$

Так как внутри формулы (11)

$$w_1 < z \leqslant v \Leftrightarrow$$
$$\Leftrightarrow (\exists w_3 \leqslant v)(\exists w_4 \leqslant v)(w_1 + w_3 + 1 = z \,\&\, z + w_4 = v),$$
$$x_i \leqslant w_1 \Leftrightarrow (\exists y_i \leqslant v)(x_i + y_i = w_2),$$

то

$$w_1 < z \leqslant v \Leftrightarrow (\exists w_3 \leqslant v)(\exists w_4 \leqslant v)(U = 0),$$
$$x_1 \leqslant w_2 \,\&\, \ldots \,\&\, x_n \leqslant w_2 \Leftrightarrow (\overset{n}{\underset{i=1}{\exists}}\ y_i \leqslant v)(V = 0), \quad (12)$$

где через $(\overset{n}{\underset{i=1}{\exists}} y_i \leqslant v)$ обозначено слово $(\exists y_1 \leqslant v) \ldots$ $\ldots (\exists y_n \leqslant v)$ и положено

$$U = (w_1 + w_3 + 1 - z)^2 + (z + w_4 - v)^2,$$
$$V = (x_1 + y_1 - w_2)^2 + \ldots + (x_n + y_n - w_2)^2.$$

При помощи соотношений (12) формулу (11) приводим к виду

$$(\exists v)\,(\forall z \leqslant v)\,(\overset{n}{\underset{i=1}{\exists}} x_i \leqslant v)\,(\overset{n}{\underset{i=1}{\exists}} y_i \leqslant v)\,(\overset{4}{\underset{i=1}{\exists}} w_1 \leqslant v)$$
$$(4\,(c\,(w_1,\ w_2) - v)^2 + U\,(F^2\,(x,\ w_1,\ z,\ x_1,\ \ldots,\ x_n) + V^2 = 0)),$$

удовлетворяющему теореме 2.

Приведенные выше доказательства теорем 1 и 2 являются конструктивными. Это значит, что, имея программу операторного алгоритма, вычисляющего частичную характеристическую функцию $f(x)$ множества M, мы в принципе можем написать многочлены F и G, удовлетворяющие требованиям теорем 1, 2. Однако эта возможность лишь принципиальная, так как если взять какую-нибудь из ныне известных программ, отвечающих нерекурсивным множествам M, и построить многочлены F, G по правилам, изложенным в доказательствах теорем 1, 2, то такие многочлены окажутся слишком сложными. Задача о нахождении более или менее простых многочленов (например, невысокой степени и с небольшим числом переменных), дающих нерекурсивное множество M, до сих пор, по-видимому, не решена.

16.3. Представимость натуральных чисел многочленами. Говорят, что число a *представимо* многочленом $F(x_1, \ldots, x_m)$, если существуют такие *натуральные* числа $a_1, \ldots, a_m$, что $F(a_1, \ldots, a_m) = a$, то есть если истинна формула

$$(\exists x_1) \ldots (\exists x_m)\,(F\,(x_1,\ \ldots,\ x_m) = a).$$

Говорят, что совокупность натуральных чисел M представляется многочленом F, если M есть совокупность всех тех *натуральных* чисел, которые представимы многочленом F. В теории чисел для ряда многочленов F исследовался вопрос о том, представимы ли этим многочленом

все натуральные числа. Например, теорема Лагранжа, которой мы уже пользовались, утверждает, что многочлен $x_1^2 + x_2^2 + x_3^2 + x_4^2$ представляет каждое натуральное число. Естественно возникает вопрос: нельзя ли указать алгоритм, посредством которого для любого заданного многочлена с целыми коэффициентами можно было бы сказать, представляет или нет этот многочлен все натуральные числа? Следуя Х. Путнаму [77], мы покажем, что этот вопрос решается отрицательно. Сначала докажем следующую теорему:

Теорема 1 (Путнам [77]). *Для каждого рекурсивно перечислимого множества* $\boldsymbol{M}$ *существует такой многочлен* $F(x, y, z_1, \ldots, z_n)$ *с целыми коэффициентами, что*

$$x \in \boldsymbol{M} \Longleftrightarrow (\exists y)(\forall z_1) \ldots (\forall z_n)(F(x, y, z_1, \ldots, z_n) \neq 0). \quad (1)$$

Согласно теореме Девиса (п. 16.2) можно найти такой многочлен $G(x, y, z, x_1, \ldots, x_m)$ с целыми коэффициентами, что

$$x \in M \Longleftrightarrow (\exists y)(\forall z \leqslant y)(\exists x_1) \ldots \\ \ldots (\exists x_m)(G(x, y, z, x_1, \ldots, x_m) = 0). \quad (2)$$

При заданных y и $z \leqslant y$ обозначим через x_{zi} натуральные числа, удовлетворяющие согласно (2) уравнению

$$G(x, y, z, x_{z1}, \ldots, x_{zm}) = 0.$$

По основному свойству функции Геделя $\boldsymbol{\Gamma}(a, t)$ системы

$$\boldsymbol{\Gamma}(a_i, z) = x_{zi} \qquad (z = 0, 1, \ldots, y)$$

имеют натуральные решения $a_1, \ldots, a_m$ и потому

$$x \in M \Longleftrightarrow (\exists y)(\exists a_1) \ldots (\exists a_m)(\forall z \leqslant y) \\ \{G(x, y, z, \boldsymbol{\Gamma}(a_1, z), \ldots, \boldsymbol{\Gamma}(a_m, z)) = 0\}$$

или

$$x \in \boldsymbol{M} \Longleftrightarrow (\exists y)(\exists a_1) \ldots (\exists a_m)(\forall z) \\ \{z > y \vee G(x, y, z, \boldsymbol{\Gamma}(a_1, z), \ldots, \boldsymbol{\Gamma}(a_m, z)) = 0\}. \quad (3)$$

Согласно теоремам 4, 5 из п. 16.1 выражение, стоящее в фигурных скобках, представляет дважды диофантов

предикат и потому равносильно формуле вида

$$(\forall z_1) \ldots (\forall z_{s-1}) (H(x, y, a_1, \ldots, a_m, z, z_1, \ldots, z_{s-1}) \neq 0), \quad (4)$$

где H — многочлен с целыми коэффициентами. Подставляя (4) в (3), получим

$$x \in M \Leftrightarrow (\exists y)(\exists a_1) \ldots (\exists a_m)(\forall z_1) \ldots (\forall z_s)$$
$$(H(x, y, a_1, \ldots, a_m, z_1, \ldots, z_s) \neq 0).$$

Вводя вместо строки $\langle y, a_1, \ldots, a_m \rangle$ ее номер $u = c(y, a_1, \ldots, a_m)$, будем иметь

$$x \in M \Leftrightarrow (\exists u)(\forall z_1) \ldots (\forall z_s)$$
$$\{H(x, c_{m+1\,1}(u), \ldots, c_{m+1\,m+1}(u), z_1, \ldots, z_s) \neq 0\}.$$

Предикат, стоящий в фигурных скобках, дважды диофантов и потому может быть заменен выражением вида

$$(\forall z_{s+1}) \ldots (\forall z_n)(F(x, u, z_1, \ldots, z_s, z_{s+1}, \ldots, z_n) \neq 0),$$

где F — подходящий многочлен с целыми коэффициентами. После указанной замены формула приобретает требуемый вид (1).

Теорема 2. *Для каждого рекурсивно перечислимого множества* M *существует такой многочлен с целыми коэффициентами* $G(x, z_1, \ldots, z_m)$, *что*

$$x \notin M \Leftrightarrow (\forall y)(\exists z_1) \ldots (\exists z_m)(G(x, z_1, \ldots, z_m) = y). \quad (5)$$

Сначала согласно теореме 1 найдем такой многочлен F, чтобы

$$x \in M \Leftrightarrow (\exists y)(\forall z_1) \ldots (\forall z_n)(F(x, y, z_1, \ldots, z_n) \neq 0),$$

и, следовательно,

$$x \notin M \Leftrightarrow (\forall y)(\exists z_1) \ldots (\exists z_n)(F(x, y, z_1, \ldots, z_n) = 0).$$

Теорема 2, очевидно, будет доказана, если будет доказана следующая формула Путнама:

$$(\exists \mathfrak{z})(F(x, y, \mathfrak{z}) = 0) \Leftrightarrow (\exists \mathfrak{z})(\exists u_1, \ldots, u_4)$$
$$(G(x, u_1, u_2, u_3, u_4, \mathfrak{z}) = y), \quad (6)$$

где $\mathfrak{z} = \langle z_1, \ldots, z_n \rangle$ и

$$G(x, u_1, \ldots, u_4, \mathfrak{z}) =$$
$$= (1 - F^2(x, u_1^2 + u_2^2 + u_3^2 + u_4^2, \mathfrak{z}))(u_1^2 + u_2^2 + u_3^2 + u_4^2 + 1) - 1. \quad (7)$$

Пусть для некоторых натуральных x, y левая часть эквивалентности (6) истинна и, следовательно, для подходящих $\mathfrak{z}$ $F(x, y, \mathfrak{z}) = 0$. По теореме Лагранжа найдутся такие натуральные u_1, u_2, u_3, u_4, что $y = u_1^2 + u_2^2 + u_3^2 + u_4^2$. Из (7) получаем

$$G(x, u_1, \ldots, u_4, \mathfrak{z}) = (1 - F^2(x, y, \mathfrak{z}))(1 + y) - 1 = y$$

и потому правая часть эквивалентности (6) истинна.

Обратно, пусть для некоторых натуральных x, y правая часть эквивалентности (6) истинна и, следовательно,

$$(1 - F^2(x, u_1^2 + u_2^2 + u_3^2 + u_4^2, \mathfrak{z}))(u_1^2 + u_2^2 + u_3^2 + u_4^2 + 1) = y + 1 \tag{8}$$

для подходящих натуральных $u_1, \ldots, u_4$. Правая часть этого равенства — число положительное. Поэтому первый множитель в левой части также должен быть положительным числом. Но это возможно только в случае, когда

$$F(x, u_1^2 + u_2^2 + u_3^2 + u_4^2, \mathfrak{z}) = 0,$$

и тогда из (8) дополнительно получаем $u_1^2 + u_2^2 + u_3^2 + u_4^2 = y$, откуда $F(x, y, \mathfrak{z}) = 0$, что и требовалось.

Следствие. *Не существует алгоритма, позволяющего для любого заданного многочлена с целыми коэффициентами узнать, представляет этот многочлен все натуральные числа или нет.*

В самом деле, берем какое-нибудь нерекурсивное рекурсивно перечислимое множество M и строим для него многочлен G, удовлетворяющий условию (5). Берем некоторое число a и рассматриваем многочлен $G(a, z_1, \ldots, z_n)$ от переменных $z_1, \ldots, z_n$. Формула (5) означает, что $a \notin M$ тогда и только тогда, когда многочлен $G(a, z_1, \ldots, z_n)$ представляет все натуральные числа. Если бы упомянутый алгоритм существовал, то при его помощи мы бы могли решать и вопросы о вхождении числа в M, что противоречит нерекурсивности множества M.

16.4. Показательные уравнения. Показательными уравнениями называются уравнения вида

$$F(x_1, \ldots, x_n, x_1^{x_1}, x_1^{x_2}, \ldots, x_n^{x_n}) = 0, \tag{1}$$

где $F(x_1, \ldots, x_n, x_{11}, x_{12}, \ldots, x_{nn})$ — некоторый многочлен от переменных x_i, x_{ij} $(i, j = 1, \ldots, n)$ с целыми рациональными коэффициентами. При этом полагают, что $x^0 = 1$ для всех целых x, в частности, $0^0 = 1$. Обычно ищут натуральные решения уравнения (1). Типичными примерами показательных уравнений могут служить уравнения

$$x^x y^y = z^z, \quad (2-x)^2 + (x^y - y)^2 = 0, \quad 2^x + 11^y = 5^z,$$

уже рассматривавшиеся в литературе.

Из определения видно, что каждое диофантово уравнение является показательным. Выше отмечалось, что проблема существования алгоритма для ответа на вопрос о разрешимости произвольно заданного диофантова уравнения пока остается открытой. Для показательных уравнений положение иное. Сравнительно недавно М. Девис, Х. Путнам и Ю. Робинсон [26] показали, что проблема разрешимости в натуральных числах произвольно заданного показательного уравнения алгоритмически неразрешима. Ниже приводится доказательство этой замечательной теоремы, принадлежащее упомянутым авторам.

Предикат $P(x_1, \ldots, x_m)$, определенный на множестве натуральных чисел, называется *показательно диофантовым*, если существует такой многочлен F от переменных x_i, x_{ij} $(i, j = 1, \ldots, n)$ с целыми коэффициентами, что

$$P(x_1, \ldots, x_m) \Leftrightarrow (\exists x_{m+1}) \ldots (\exists x_n) \\ (F(x_1, \ldots, x_n, x_1^{x_1}, x_1^{x_2}, \ldots, x_n^{x_n}) = 0).$$

Числовая функция $f(x_1, \ldots, x_m)$ называется *показательно диофантовой*, если отношение $f(x_1, \ldots, x_m) = x_{m+1}$ показательно диофантово.

Отсюда следует, что все диофантовы отношения и функции являются показательно диофантовыми.

Теорема 1. *Если предикаты $P(x_1, \ldots, x_n)$, $Q(x_1, \ldots, x_n)$ и функции $f(x_1, \ldots, x_n)$, $g_1(x_1, \ldots, x_m), \ldots, g_n(x_1, \ldots, x_m)$ показательно диофантовы, то функция*

$$f(g_1(x_1, \ldots, x_m), \ldots, g_n(x_1, \ldots, x_m))$$

и предикаты

$$P(x_1, \ldots, x_n) \vee Q(x_1, \ldots, x_n),$$
$$P(x_1, \ldots, x_n) \,\&\, Q(x_1, \ldots, x_n),$$
$$(\exists x_n) P(x_1, \ldots, x_n),$$
$$P(g_1(x_1, \ldots, x_m), \ldots, g_n(x_1, \ldots, x_m)),$$
$$g_1(x_1, \ldots, x_m) = g_2(x_1, \ldots, x_m),$$
$$g_1(x_1, \ldots, x_m) < g_2(x_1, \ldots, x_m)$$

также показательно диофантовы.

Доказательство такое же, как и доказательство соответствующей теоремы о диофантовых функциях и предикатах в п. 16.1.

Из теоремы 1, например, вытекает, что показательно диофантовыми являются следующие функции и отношения:

$$2^{x^2},\ (2^x+1)^x,\ (3^x+1)^y-2^z,\ 3^x<2^y.$$

Мы хотим, следуя Девису, Путнаму и Робинсон, доказать, что показательно диофантовыми являются все частично рекурсивные функции и рекурсивно перечислимые предикаты.

Лемма 1 (Ю. Робинсон [88]). *Функция*

$$\binom{n}{k} = \frac{n(n-1)\ldots(n-k+1)}{k!}$$

от переменных n, k показательно диофантова.

Пусть $0 < k \leqslant n$, тогда

$$2^{nk}(1+2^{-n})^n =$$
$$= 2^{nk}\sum_{i=0}^{n}\binom{n}{i}2^{-ni} < \sum_{i=0}^{k}\binom{n}{i}2^{n(k-i)} + (2^n-1)/2^n.$$

Следовательно,

$$[2^{nk}(1+2^{-n})^n] = \sum_{i=0}^{k}\binom{n}{i}2^{n(k-i)},$$

где $[x]$ — наибольшее целое число, не превосходящее x.

При тех же предположениях $0 < k \leqslant n$ аналогичным путем получаем формулу

$$2^n [2^{n(k-1)} (1+2^{-n})^n] = \sum_{i=0}^{k-1} \binom{n}{i} 2^{n(k-i)}$$

и, следовательно,

$$\binom{n}{k} = [2^{nk} (1+2^{-n})^n] - 2^n [2^{n(k-1)} (1+2^{-n})^n]. \qquad (1)$$

Из справедливости формулы (1) для $0 < k \leqslant n$ вытекает, что для произвольных k, n имеет место эквивалентность

$$y = \binom{n}{k} \Leftrightarrow (n < k \,\&\, k > 0 \,\&\, y = 0) \vee (k = 0 \,\&\, y = 1) \vee$$
$$\vee \left(n \geqslant k \,\&\, k > 0 \,\&\, y = \left[\frac{2^{nk} (2^n+1)^n}{2^{n^2}} \right] - \right.$$
$$\left. - 2^n \left[\frac{2^{n(k-1)} (2^n+1)^n}{2^{n^2}} \right] \right).$$

Все отношения, стоящие здесь справа от знака $\Leftrightarrow$, являются показательно диофантовыми. Поэтому и функция $\binom{n}{k}$ показательно диофантова.

Лемма 2 (Ю. Робинсон [88]). *Функция $y = x!$ показательно диофантова.*

Докажем формулу

$$r > (2x)^{x+1} \Rightarrow x! = \left[r^x \Big/ \binom{r}{x} \right]. \qquad (2)$$

При $x = 0$ эта формула очевидна. Пусть $x > 0$. Тогда

$$1/(1-\theta) < 1 + 2\theta \text{ для } 0 < \theta < \frac{1}{2},$$

$$(1+\theta)^x < 1 + 2^x \theta \text{ для } 0 < \theta < 1,$$

$$x! < r^x \Big/ \binom{r}{x} = x! \Big/ \left(1 - \frac{1}{r}\right)\left(1 - \frac{2}{r}\right) \ldots \left(1 - \frac{x-1}{r}\right) <$$
$$< x! \Big/ \left(1 - \frac{x}{r}\right)^x < x! \left(1 + \frac{2x}{r}\right)^x <$$
$$< x! \left(1 + 2^x \frac{2x}{r}\right) < x! + 1.$$

Полагая в формуле (2) $r=(2(x+1))^{x+1}$, получим

$$x! = \left[(2(x+1))^{x(x+1)} \Big/ \binom{(2(x+1))^{x+1}}{x} \right].$$

На основании теоремы 1 и леммы 1 заключаем, что функция $x!$ показательно диофантова.

Лемма 3. *Отношение между числами x, y, p, q, k, выражаемое формулой*

$$\frac{x}{y} = \binom{p/q}{k} \,\&\, p > qk,$$

является показательно диофантовым.

Пусть $\alpha = p/q$, $\alpha > k$. По формуле Тэйлора с остаточным членом в форме Лагранжа имеем

$$a^{2k+1}(1+a^{-2})^{\alpha} = \sum_{j=0}^{k} \binom{\alpha}{j} a^{2k-2j+1} + $$
$$+ \binom{\alpha}{k+1} a^{-1} (1+\theta a^{-2})^{\alpha-k-1} =$$
$$= \sum_{j=0}^{k} \binom{\alpha}{j} a^{2k-2j+1} + \theta' \alpha^{k+1} a^{-1} 2^{\alpha-1}, \qquad (3)$$

где $0 \leqslant \theta \leqslant 1$, $0 \leqslant \theta' \leqslant 1$. Положим

$$S_k(a) = \sum_{j=0}^{k} \binom{\alpha}{j} a^{2k-2j+1} \qquad (4)$$

и будем искать такое целое a, чтобы

$$S_k(a) = [a^{2k+1}(1+a^{-2})^{\alpha}], \qquad (5)$$
$$S_{k-1}(a) = [a^{2k-1}(1+a^{-2})^{\alpha}]. \qquad (6)$$

Допустим, что a делится на $q^k k!$. Тогда $S_k(a)$ и $S_{k-1}(a)$ будут целыми числами (см. формулу (4)). С другой стороны, если $a > 2^{p-1}p^{k+1}$, то остаточные члены в формуле (3) и в аналогичной формуле для $k-1$ будут меньше чем 1. Таким образом, формулы (5) и (6) истинны для $a = 2^p p^{k+1} q^k k!$. Но из (4) следует

$$\binom{p/q}{k} = a^{-1} S_k(a) - a S_{k-1}(a).$$

Все функции, встречающиеся в этой формуле справа от знака равенства, являются показательно диофантовыми и лемма 3 доказана.

Лемма 4. *Функция* $\psi(a, b, y)$, *определенная равенством*

$$\psi(a, b, y) = \prod_{i=0}^{y} (a + bi),$$

является показательно диофантовой.

Это непосредственное следствие предыдущих лемм и формулы

$$\prod_{i=0}^{y} (a + bi) = \binom{a/b + y}{y} ab^y y!.$$

Перейдем к доказательству основного результата.

Теорема 2 (Девис — Путнам — Робинсон [26]). *Каждый рекурсивно перечислимый предикат показательно диофантов.*

Достаточно рассмотреть случай, когда предикат одноместен и потому эквивалентен отношению $x \in \boldsymbol{M}$, где $\boldsymbol{M}$ — рекурсивно перечислимое множество чисел. Согласно теореме Девиса (п. 16.2) существует такой многочлен $F(x, y, z, x_1, \ldots, x_m)$ с целыми коэффициентами, что

$$x \in \boldsymbol{M} \Leftrightarrow (\exists y)(\forall z \leqslant y)(\exists x_1 \leqslant y) \ldots (\exists x_m \leqslant y)(G = 0).$$

Поэтому остается лишь доказать, что отношение

$$(\forall z \leqslant y)(\exists x_1 \leqslant y) \ldots (\exists x_m \leqslant y)(G(x, y, z, x_1, \ldots, x_m) = 0)$$

показательно диофантово.

Пусть u — степень G, s — сумма абсолютных величин коэффициентов G. Полагая

$$H(x, y) = (s + 1)(x + 1)^u (y + 1)^u,$$

будем иметь

$$H(x, y) \geqslant y,$$

$$(\forall z \leqslant y)(\forall x_1 \leqslant y) \ldots (\forall x_m \leqslant y)(|G(x, y, z, x_1, \ldots, x_m)| \leqslant H(x, y)).$$

Докажем, что имеет место эквивалентность

$$(\forall z \leqslant y)\,(\exists x_1 \leqslant y) \ldots (\exists x_m \leqslant y)\,(G(x, y, z, x_1, \ldots, x_m) = 0) \Leftrightarrow$$

$$\Leftrightarrow (\exists c)\,(\exists t)\,(\exists a_1) \ldots (\exists a_m)\,\{t = H(x, y)!\,\&$$

$$\&\, 1 + (1+c)\,t = \prod_{i=0}^{y+1} (1+it)\,\&\,(1+(1+c)\,t) \mid G(x, y, c, a_1, \ldots, a_m)\,\&$$

$$\&\,(1+(1+c)\,t) \mid \prod_{i \leqslant y} (a_1 - i)\,\&\ldots\&\,(1+(1+c)\,t) \mid \prod_{i \leqslant y} (a_m - i)\}. \quad (7)$$

Пусть для некоторых x, y правая часть этой эквивалентности истинна. Покажем, что левая часть также истинна. Рассмотрим какое-нибудь натуральное z, не превосходящее y, и пусть p_z — простое число, делящее $1 + (z + 1)\,t$. Тогда, полагая

$$\operatorname{rest}(a_j, p_z) = x_{zj} \qquad (j = 1, \ldots, m),$$

достаточно показать, что

$$x_{zj} \leqslant y \quad (8)$$

и

$$G(x, y, z, x_{z1}, \ldots, x_{zm}) = 0. \quad (9)$$

Согласно (7) p_z делит $\prod_{i \leqslant y} (a_j - i)$ и потому $p_z \mid a_j - i$ для подходящего i, $0 \leqslant i \leqslant y$. Следовательно,

$$\operatorname{rest}(a_j, p_z) = \operatorname{rest}(i, p_z) \leqslant y$$

и (8) доказано.

Далее, из $(p_z, t) = 1$ и $t = H(x, y)!$ следует $p_z > H(x, y)$. Так как $x_{zj} \leqslant y$, $z \leqslant y$, то

$$|G(x, y, z, x_{z1}, \ldots, x_{zm})| \leqslant H(x, y) < p_z. \quad (10)$$

Но $1 + (c + 1)\,t \equiv 0 \pmod{1 + (z + 1)\,t}$, поэтому $c \equiv z \pmod{1 + (z+1)\,t}$ и $c \equiv z \pmod{p_z}$. Так как

$$x_{zi} \equiv a_i \pmod{p_z} \text{ и } (1 + (1+c)\,t) \mid G(x, y, c, a_1, \ldots, a_m),$$

то

$$G(x, y, z, x_{z1}, \ldots, x_{zm}) \equiv G(x, y, c, a_1, \ldots, a_m) \equiv 0 \pmod{p_z}. \quad (11)$$

Из (10) и (11) получаем (9).

Обратно, предположим, что для фиксированных x, y левая часть эквивалентности (7) истинная и, следовательно, для каждого $z = 0, 1, \ldots, y$ существуют числа $x_{z1}, \ldots$ $\ldots, x_{zm} \leqslant y$ такие, что

$$G(x, y, z, x_{z1}, \ldots, x_{zm}) = 0.$$

Полагаем $H(x, y)! = t$ и определяем c из условия

$$1 + (c+1)\,t = \prod_{i=0}^{y+1} (1 + it).$$

Тогда для каждых $i, j \leqslant y$, $i \neq j$, числа $1 + (i+1)\,t$ и $1 + (j+1)\,t$ взаимно просты.

Действительно, если простое число p делит одновременно указанные числа, то p делит и $(j - i)\,t$. Но

$$|i - j| \leqslant y \leqslant H(x, y), \quad t = H(x, y)!,$$

поэтому p делит и число t, что невозможно, так как p не может делить одновременно t и $1 + (i+1)\,t$.

Итак, числа $1 + (i+1)\,t$ для $i = 0, 1, \ldots, y$ попарно взаимно простые. Согласно известной теореме об остатках (см. п. 3.3) отсюда следует существование натуральных чисел $a_1, \ldots, a_m$, удовлетворяющих сравнениям

$$a_i \equiv x_{zi} \,(\mathrm{mod}\,(1 + (z+1)\,t)) \quad (z = 0, 1, \ldots, y;\ i = 1, 2, \ldots, m). \tag{12}$$

Как и прежде, $c \equiv z \,(\mathrm{mod}\,(1 + (z+1)\,t))$, поэтому

$$G(x, y, c, a_1, \ldots, a_m) \equiv G(x, y, z, x_{z1}, \ldots, x_{zm}) \equiv$$
$$\equiv 0 \,(\mathrm{mod}\,(1 + (z+1)\,t))$$

для $z = 0, \ldots, y$. Так как модули взаимно просты, то

$$1 + (c+1)\,t \mid G(x, y, c, a_1, \ldots, a_m).$$

Аналогично, из (12) имеем

$$1 + (z+1)\,t \mid a_i - x_{zi}, \quad x_{zi} \leqslant y,$$

и потому

$$1 + (z+1)\,t \mid \prod_{j=0}^{y} (a_i - j)$$

для каждого $z \leqslant y$. Так как эти делители попарно взаимно просты, то

$$\prod_{i=0}^{y+1} (1+it) \mid \prod_{j=0}^{y} (a_i - j)$$

и потому правая часть эквивалентности (7) истинна.

Для окончания доказательства теоремы остается лишь проверить, что все отношения, находящиеся в правой части эквивалентности (7), показательно диофантовы. Показательная диофантность первых трех отношений, участвующих в этой части, была установлена выше. Что касается остальных отношений, то они также показательно диофантовы, так как

$$1+(1+c)t \mid \prod_{j=0}^{y} (a_i - j) \Leftrightarrow a_i \leqslant y \vee$$

$$\vee (\exists u)\, (a_i = y+u+1 \;\&\; 1+(1+c)\,t \mid \prod_{j=0}^{y} (u+1+j)).$$

Теорема доказана.

Дополнения и примеры

1. Для любого целого $w \geqslant 2$ совокупность натуральных чисел, не являющихся степенями w, диофантова. Будет ли для какого-нибудь $w \geqslant 2$ совокупность всех степеней w диофантовой — пока не решенная проблема (Ю. Робинсон [88]).

2. Для каждого $n \geqslant 1$ отношение

$$P(a_0, a_1, \ldots, a_n) \Leftrightarrow (\exists x)\,(a_0 x^n + \ldots + a_{n-1}x + a_n = 0)$$

дважды диофантово. В частности, дважды диофантово отношение $(\exists y)\,(F(x, y) = 0)$, где F — многочлен от переменных x, y, имеющий целые коэффициенты (А. Тарский, см. [88]).

3. n-я суперстепень x, символически $x * n$, определяется рекурсией

$$x * 0 = 1, \quad x * (n+1) = x^{x*n}.$$

Ю. Робинсон [88] показала, что если существует диофантово отношение $P(x, y)$, удовлетворяющее требованиям

$$(\exists n)\,(\forall xy)\,(P(x, y) \rightarrow y < x * n),$$

$$\forall n)\,(\exists xy)\,(P(x, y) \;\&\; y \geqslant x^n),$$

то каждое рекурсивно перечислимое отношение диофантово.

4. В теореме 2 из п. 16.2 число m не фиксировано. Р. Робинсон [90] дал другое доказательство, из которого следует, что в упомянутой теореме можно взять $m = 4$, то есть что для каждого рекурсивно перечислимого множества $\boldsymbol{M}$ существует многочлен F от *семи* переменных с целыми коэффициентами, для которого

$$x \in \boldsymbol{M} \Longleftrightarrow (\exists y)(\forall z \leqslant y)(\exists x_1 \leqslant y)\ldots$$
$$\ldots(\exists x_4 \leqslant y)(F(x, y, z, x_1, x_2, x_3, x_4) = 0).$$

5. Не существует алгоритма, позволяющего для любого многочлена с целыми коэффициентами сказать, представляет ли он все натуральные числа, начиная с некоторого, или нет (Х. Путнам [77]).

6. Для каждого рекурсивно перечислимого множества $\boldsymbol{M}$ существует такой многочлен F с целыми коэффициентами, что

$$x \in \boldsymbol{M} \Longleftrightarrow (\exists y \geqslant x)(\forall x_1 \leqslant y)\ldots(\forall x_n \leqslant y)(F(x, y, x_1, \ldots, x_n) = 0)$$

(Х. Путнам [77]).

7. 10-я проблема Гильберта неразрешима, если для каждого целого положительного k существуют a, b, x, y такие, что $ax^3 - by^3 = 1$ и хотя бы одно из чисел x, y больше, чем $(a+b)^k$ (Девис и Путнам [27], Девис [24]).

8. В кольце $N[\zeta]$ всех многочленов от одной переменной ζ с целыми коэффициентами каждое рекурсивно перечислимое множество $\boldsymbol{M}$ положительных чисел диофантово. Это означает, что существует многочлен $F(X, Y_1, \ldots, Y_n)$ с коэффициентами из $N(\zeta)$ такой, что

$$a \in \boldsymbol{M} \Longleftrightarrow (\exists Y_1)\ldots(\exists Y_n)(F(a, Y_1, \ldots, Y_n) = 0),$$

где кванторы относятся к совокупности всех элементов кольца $N(\zeta)$ (Девис и Путнам [27]).

ЛИТЕРАТУРА

1. Адян С. И., О проблеме делимости в полугруппах, Докл. АН СССР **103**, № 5 (1955), 747—750.
2. Адян С. И., Неразрешимость некоторых алгебраических проблем теории групп, Тр. Моск. матем. о-ва **6** (1957), 231—298.
3. Адян С. И., К проблеме тождества в ассоциативных системах специального вида, Докл. АН СССР **135**, № 6 (1960), 1297—1300.
4. Аккерман В. (Ackermann W.), Zum Hilbertschen Aufbau der reelen Zahlen, Math. Ann. **99** (1928), 118—133.
5. Бардзинь Я. М., Об одном классе машин Тьюринга (машины Минского), Алгебра и логика (семинар) **1**, № 6 (1962), 42—51.
6. Баумслаг Г., Бун В. и Нейман Б. (Baumslag G., Boone W. and Neumann B.), Some unsolvable problems about elements and subgroups of groups, Math. scand. **7** (1959), 191—201.
7. Берецки И. (Berecki I.), Nem-elemi rekurziv függveny létezésc, Comptes Rendus du Premier Congrés des Mathématiciens Hongrois, 1950, 409—417.
8. Бриттон Дж. (Britton J. L.), The word problem, Ann. Math. **77**, № 1 (1963), 16—32.
9. Бун В. (Boone W.), The word problem, Ann. Math. **70**, № 2 (1959), 207—265.
10. Бюхи Дж. (Büchi J. R.), Regular canonical systems, Archiv für Math. Logik und Grundlagenf. **6**, № 3—4 (1964), 91—111.
11. Ван Хао (Wang Hao), A Variant to Turings Theory of Computing Machines, J. Assoc. Comp. Mach. **4**, № 1 (1957), 63—92.
12. Ван Хао (Wang Hao), Tag Systems and Lag Systems, Math. Ann. **152**, № 1 (1963), 65—74.
13. Ватанабе С. (Watanabe C.), 5-symbol 8-state and 5-symbol 6-state universal Turing machines, J. Assoc. Comp. Mach. **8**, № 4 (1964), 476—483.
[Русский перевод: Универсальные машины Тьюринга с 5 символами 8 состояниями и 5 символами 6 состояниями, Кибернетический сборник **6** (1963), 80—90].
14. Гедель К. (Gödel Kurt), Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme 1, Monatsh. Math. u. Phys. **38** (1931), 173—198.

15. Гильберт Д. (Hilbert D.), Über das Unendliche, Math. Ann. **95** (1926), 161—190.
16. Гильберт Д. (Hilbert D.), Mathematische Probleme, Nachr. K. Ges. Wiss. Göttingen, Math.— Phys. Kl., 1900, 253—297.
17. Гильберт Д. и Бернайс П. (Hilbert D. und Bernays P.), Grundlagen der Mathematik, Berlin, Bd. 1, 1934, Bd. 2, 1939.
18. Гинзбург С. и Роуз Дж. (Ginsburg S. and Rose G. F.), A comparison of the work done by generalized sequential machine and Turing machines, Trans. Amer. Math. Soc. **103** (1962), 394—402.
19. Глушков В. М., Теория алгоритмов, Киев, 1961, стр. 1—168.
20. Глушков В. М., Синтез цифровых автоматов, Физматгиз, 1962.
21. Гжегорчик А. (Grzegorczyk A.), Some classes of recursive functions, Rozpr. mat., Warsaw, 1953.
22. Гуревич Ю. Ш., Элементарные свойства упорядоченных абелевых групп, Алгебра и логика (семинар) **3**, № 1 (1964), 5—39.
23. Девис М. (Davis M.), Computability and unsolvability, N. Y., 1958.
24. Девис М. (Davis M.), Unsolvable problems, A review, Proc. Sympos. Math. Theory of Automata, N. Y., 1962.
25. Девис М. (Davis M.), Extensions and corollaries of recent work on Hilbert's tenth problem, Illinois J. Math. **7**, № 2 (1963), 246—250.
26. Девис М., Путнам Х. и Робинсон Ю. (Davis M., Putnam H. and Robinson J.), The decision problem for exponential Diophantine equations, Ann. Math. **74** (1961), 425—436.
(Русский перевод см. Математика, № 5 (1964)).
27. Девис М., Путнам Х. (Davis M., Putnam H.), Diophantine sets over polinomial rings, Illinois J. Math. **7**, № 1 (1963), 251—256.
28. Деккер Дж. (Decker J. C.), Two notes on recursively enumerable sets, Proc. Amer. Math. Soc. **4** (1953), 495—501.
29. Деккер Дж. (Decker J. C.), Productive sets, Trans. Amer. Math. Soc. **73** (1955), 129—149.
30. Деккер Дж. и Майхил Дж. (Decker J. C. and Myhill J.), Some theorems on classes of recursively enumerable sets, Trans. Amer. Math. Soc. **89** (1958), 25—59.
31. Деккер Дж. и Майхил Дж. (Decker J. and Myhill J.), Retraceable sets, Canad. J. Math. **10** (1958), 357—373.
32. Деккер Дж. и Майхил Дж. (Decker J. and Myhill J.), Recursive equivalence types, Univ. of California Publ. in Math., New Ser. **3**, № 3 (1960), 67—214.
33. Детловс В. К., Нормальные алгоритмы и рекурсивные функции, Докл. АН СССР **90** (1953), 723—725.

34. Ершов Ю. Л., Лавров И. А., Тайманов А. Д., Тайцлин М. А., Элементарные теории, Успехи матем. наук **20**, № 4 (1965).
35. Ершов Ю. Л., Неразрешимость теорий симметрических и простых конечных групп, Докл. АН СССР **158**, № 4 (1964), 777—779.
36. Ейтс С. (Yates C. E.), Recursively enumerable sets and retracing functions, Z. math. Logik und Grundl. Math. **8** (1962), 331—345.
37. Зыкин Г. П., Замечание об одной теореме Хао Вана, Алгебра и логика (семинар) **2**, № 1 (1963), 33—35.
38. Кальмар Л. (Kalmar L.), Egyszerü példa eldönthetetlen aritmetikai problémára, Mathematikai és Fizikai Lapok **50** (1943), 1—23.
39. Клив Дж. (Cleave J. P.), Creative functions, Z. math. Logik und Grundl. Math. **7** (1961), 205—212.
40. Клив Дж. (Cleave J. P.), A hierarchy of primitive recursive functions, Z. math. Logik und Grundl. Math. **9**, № 4 (1963), 331—345.
41. Клини С. (Kleene S. C.), General recursive functions of natural numbers, Math. Ann. **112** (1936), 727—742.
42. Клини С. (Kleene S. C.), λ-definability and Recursiveness, Duke Math. J. **2** (1936), 340—353.
43. Клини С. (Kleene S. C.), Introduction to Metamathematics, Princeton, N. J., 1952.
[Русский перевод: Введение в метаматематику, ИЛ, 1957.]
44. Кокк и Минский М. (Cocke and Minsky M. I.), Universality of p=2 Tag systems, A. I. Memo. № 52, Cambridge, Mass (1963).
45. Колмогоров А. Н. и Успенский В. А., К определению алгоритма, Успехи матем. наук **13**, № 4 (1958), 3—28.
46. Кузнецов А. В., О примитивно рекурсивных функциях большого размаха, Докл. АН СССР **71**, № 2 (1950), 233—236.
47. Лембек И. (Lambeck I.), How to program an infinite abacus, Canad. Math. Bull. **4**, № 3 (1961).
48. Лю С. (Liu S. C.), Proof of the conjecture of Routlege, Proc. Amer. Math. Soc. **11**, № 6 (1960), 967—969.
49. Лю С. (Liu S. C.), A theorem on general recursive functions, Proc. Amer. Math. Soc. **11**, № 2 (1960), 184—187.
50. Мальцев А. И., Эффективная неотделимость множеств тождественно истинных и конечно опровержимых формул некоторых элементарных теорий, Докл. АН СССР **139**, № 4 (1961), 802—804.
51. Мальцев А. И., Конструктивные алгебры, 1, Успехи матем. наук **16**, № 3 (1961), 3—60.
52. Мальцев А. И., Полно нумерованные множества, Алгебра и логика (семинар) **2**, № 2 (1963), 4—30.
53. Мальцев А. И., К теории вычислимых семейств объектов, Алгебра и логика (семинар) **3**, № 4 (1964), 5—31.

54. Марков А. А., О представлении рекурсивных функций, Изв. АН СССР, серия матем. **13**, № 5 (1949), 417—424.
55. Марков А. А., Теория алгорифмов, Тр. матем. ин-та АН СССР, т. 42, 1954.
56. Марков А. А., К проблеме представимости матриц, Z. math. Logik und Grundl. Math. **4**, № 2 (1958), 157—168.
57. Майхил Дж. (Myhill J.), Creative sets, Z. math. Logik und Grundl. Math. **1** (1955), 97—108.
58. Майхил Дж. (Myhill J.), A stumbling block in constructive Mathematics, J. Symbolic Logic **18** (1953), 190—191.
59. Медведев Ю. Т., О неизоморфных рекурсивно перечислимых множествах, Докл. АН СССР **102**, № 2 (1955), 211—214.
60. Минский М. (Minsky M. L.), Recursive unsolvability of Post's problem of «TAG» and topics in theory of Turing machines, Ann. Math. **74** (1961), 437—455.
61. Михайлова К. А., Проблема вхождения для прямых произведений групп, Докл. АН СССР **119**, № 6 (1958), 1103—1105.
62. Мостовский А. (Mostowski A.), A formula with no recursively enumerable model, Fundam. math. **41**, № 1 (1955), 125—140.
63. Мучник А. А., Об отделимости рекурсивно перечислимых множеств, Докл. АН СССР **109**, № 1 (1956), 29—32.
64. Мучник А. А., Изоморфизм систем рекурсивно перечислимых множеств с эффективными свойствами, Тр. Моск. матем. о-ва **7** (1958), 407—412.
65. Нагорный Н. М., К усилению теоремы приведения теории нормальных алгоритмов, Докл. АН СССР **90**, № 3 (1953), 341—342.
66. Новиков П. С., Об алгоритмической неразрешимости проблемы тождества слов в теории групп, Тр. матем. ин-та АН СССР, т. 44, 1955, стр. 1—144.
67. Петер Р. (Péter Rózsa), Rekursive Funktionen, Budapest 1951, [Русский перевод: Рекурсивные функции, ИЛ, 1954.]
68. Петер Р. (Péter Rózsa), Programmierung und partiellrekursive Funktionen, Acta math. hung. **14** (1963), 373—401.
69. Поляков Е. А., Алгебры рекурсивных функций, Алгебра и логика (семинар) **3**, № 1 (1964), 41—55.
70. Пост Э. (Post E. L.), Finite combinatory processes—Formulation, J. Symbolic Logic **1** (1936), 103—105.
71. Пост Э. (Post E. L.), Formal reductions of the combinatorial decision problem, Amer. J. Math. **65** (1943), 197—215.
72. Пост Э. (Post E. L.), A variant of a recursively unsolvable problem, Bull. Amer. Math. Soc. **52**, № 4 (1946), 264—268.
73. Пост Э. (Post E. L.), Recursive unsolvability of a problem of Thue, J. Symbolic Logic **12**, № 1 (1947), 1—11.
74. Пост Э. (Post E. L.), Recursively enumerable sets of positive integers and their decision problems, Bull. Amer. Math. Soc. **50** (1944), 284—316.

75. Пур-Эль М. (Pour-El M. B.), Gödel numberings versus Friedberg numberings, Proc. Amer. Math. Soc. **15**, № 2 (1964), 252—255.
76. Пур-Эль М. и Ховард В. (Pour-El M. B. and Howard W. A.), A structural criterion for recursive enumeration without repetition, Z. math. Logik und Grundl. Math. **10**, № 2 (1964), 105—114.
77. Путнам Х. (Putnam H.), An unsolvable problem in number theory, J. Symbolic Logic **25**, № 3 (1960), 220—232.
78. Путнам Х. (Putnam H.), On hierarchies and systems of notations, Proc. Amer. Math. Soc. **15**, № 1 (1964), 44—50.
79. Рабин М. (Rabin M. O.), Recursive unsolvability of group theoretic problems, Ann. Math. **67** (1958), 172—194.
80. Рабин М. (Rabin M. O.), Computable algebra, general theory and theory of computable fields, Trans. Amer. Math. Soc. **95**, № 2 (1960), 341—360.
81. Райс Х. (Rice H. G.), Classes of recursively enumerable sets and their decision problems, Trans. Amer. Math. Soc. **74** (1953), 358—366.
82. Райс Х. (Rice H. G.), On completely recursively enumerable classes and their key arrays, J. Symbolic Logic **21** (1956), 304—308.
83. Райс Х. (Rice H. G.), Recursive and recursively enumerable orders, Trans. Amer. Math. Soc. **83** (1956), 277—300.
84. Раутледж Н. (Routlege N. A.), Ordinal recursion, Proc. Cambridge Phil. Soc. **49** (1953), 175—182.
85. Ричи Р. (Ritchie R. W.), Classes of predictable computable functions, Trans. Amer. Math. Soc. **106** (1963), 109—163.
86. Робинсон А. (Robinson A.), Introduction to Model Theory and to the Metamathematics of Algebra, Amsterdam, 1963.
87. Робинсон Ю. (Robinson Julia), General recursive functions, Proc. Amer. Math. Soc. **1** (1950), 703—718.
88. Робинсон Ю. (Robinson Julia), Existential Definability in Arithmetic, Trans. Amer. Math. Soc. **72** (1952), 437—439.
89. Робинсон Р. (Robinson R. M.), Primitive recursive functions, Bull. Amer. Math. Soc. **53** (1947), 925—942.
90. Робинсон Р. (Robinson R. M.), Arithmetical representation of recursively enumerable sets, J. Symbolic Logic **21** (1956), 162—186.
91. Роджерс Х. (Rogers Hartley, Jr.), Gödel numberings of partial recursive functions, J. Symbolic Logic **23** (1958), 331—341.
92. Сакс Дж. (Sacks G. E.), A simple set which is not effectively simple, Proc. Amer. Math. Soc. **15**, № 1 (1964), 51—55.
93. Сколем Т. (Skolem Th.), Remarks on recursive functions and relations, Det Kongelige Norske Videnskabers Selskab. Forhandlinger **17** (1944), 89—92.

94. Сколем Т. (Skolem Th.), Some remarks on recursive arithmetic, Det Kongelige Norske Videnskabers Selskab, Forhandlinger **17** (1944), 103—106.
95. Сколем Т. (Skolem Th.), A theorem on recursively enumerable sets, Abstr. of short comm. Int. Congress Math., 1962, Stockholm, p. 11.
96. Тарский А. (Tarski A.), Der Wahrheitsbegriff in den formalisierten Sprachen, Studia Philosophica **1** (1936), 261—405.
97. Тарский А., Мостовский А. и Робинсон Р. (Tarski A., Mostowski A. and Robinson R. M.), Undecidable Theories, Amsterdam, 1953.
98. Тарский А. и Мак-Кинси Дж. (Tarski A. and McKinsey J. C. C.), A Decision Method for Elementary Algebra and Geometry, 2-nd Ed, Berkeley, 1951.
99. Трахтенброт Б. А., Алгоритмы и машинное решение задач, Физматгиз, 1960.
100. Триттер А. (Tritter A.), Universal Turing machine with 4 symbols and 6 states (см. Ван Хао [12]).
101. Тьюринг А. М. (Turing A. M.), On computable numbers with an application to the Entscheidungsproblem, Proc. London Math. Soc. (2) **42** (1937), 230—265 (Correction: **43** (1947), 544—546).
102. Тьюринг А. М. (Turing A. M.), Computability and λ-definability, J. Symbolic Logic **2** (1937), 153—163.
103. Успенский В. А., Системы перечислимых множеств и их нумераций, Докл. АН СССР **105**, № 6 (1955), 1155—1158.
104. Успенский В. А., Несколько замечаний о перечислимых множествах, Z. math. Logik und Grundlagen Math. **3** (1957), 157—170.
105. Успенский В. А., Лекции о вычислимых функциях, Физматгиз, 1960.
106. Феферман С. (Feferman S.), Classification of recursive functions by means of hierarchies, Trans. Amer. Math. Soc. **104** (1962), 101—122.
107. Фишер П. (Fischer P. O.), A note on bounded-truth-table reducibility, Proc. Amer. Math. Soc. **14** (1963), 875—877.
108. Фридберг Р. (Friedberg R. M.), Three theorems on recursive functions: I. Decomposition. II. Maximal sets. III. Enumeration without duplication, J. Symbolic Logic. **23** (1958), 309—318.
109. Фридберг Р. и Роджерс Х. (Friedberg R. and Rogers H., Jr.), Reducibility and completeness for sets of integers, J. Symbolic Logic **24** (1959), 117—125.
110. Хигман Г. (Higman G.), Subgroups of finitely presented groups, Proc. Roy. Soc. **262**, № 1311 (1961), 455—475.
111. Холл М. (Hall M., Jr.), The word problem for semi-groups with two generators, J. Symbolic Logic **14** (1949), 115—118.
112. Цейтин Г. С., Ассоциативное исчисление с неразрешимой проблемой эквивалентности, Тр. матем. ин-та АН СССР **52** (1958), 172—189.

113. Цейтин Г. С., Изложение теории алгоритмов на языке бинарных отношений, 5-й Всес. коллоквиум по общей алгебре, Резюме сообщений и докладов, Новосибирск, Институт математики, 1963, стр. 55.
114. Чёрч А. (Church Alonzo), An unsolvable problem of elementary number theory, Amer. J. Math. **58** (1936), 345—363.
115. Чёрч А. (Church Alonzo), A note on the Entscheidungsproblem, J. Symbolic Logic **1** (1936), 40—41 (Correction, ibid., pp. 101—102).
116. Чёрч А. (Church Alonzo), The Calculi of Lambda-conversion, Princeton, N. J., 1941.
117. Шеннон С. (Shannon C. E.), A universal Turing machine with two internal states, Automata Studies, Princeton, 1956.
[Русский перевод: см. в сборнике «Автоматы», ИЛ, 1956.]
118. Шефердсон Дж. и Стургис Х. (Shepherdson J. C. and Sturgis H. E.), Computability of recursive functions, J. Assoc. Comput. Machinery **4** (1963), 217—255.
119. Шмелева В. (Szmielew W.), Elementary properties of Abelian groups, Fundam. Math. **41**, № 2 (1955), 203—271.
120. Шмульян Р. (Smullian R. M.), On Post's canonical systems, J. Symbolic Logic **27** (1962), 55—57.
121. Шмульян Р. (Smullian R. M.), Theory of formal systems, Ann. Math. Studies, 1961.

УКАЗАТЕЛЬ ИМЕН

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

ОСНОВНЫЕ ОБОЗНАЧЕНИЯ

Ниже для ряда основных обозначений указываются страницы книги, где впервые вводятся или определяются соответствующие понятия или объекты. Кратко поясняются лишь обозначения некоторых общих математических понятий.

$\{a, b, \ldots, c\}$ — множество, состоящее из элементов $a, b, \ldots, c$;

$\{x \mid A(x)\}$ — множество тех значений x, которые удовлетворяют условию $A(x)$ (т. е. для которых $A(x)$ истинно);

$x \in \boldsymbol{A}$ — элемент x принадлежит множеству $\boldsymbol{A}$;

$x \notin \boldsymbol{A}$ — элемент x не принадлежит множеству $\boldsymbol{A}$;

$\boldsymbol{A} \subseteq \boldsymbol{B}$ — $\boldsymbol{A}$ есть часть $\boldsymbol{B}$;

$\boldsymbol{A} \nsubseteq \boldsymbol{B}$ — $\boldsymbol{A}$ не есть часть $\boldsymbol{B}$;

$\boldsymbol{A} \cup \boldsymbol{B}$ — объединение множеств $\boldsymbol{A}$ и $\boldsymbol{B}$;

$\boldsymbol{A} \cap \boldsymbol{B}$ — пересечение множеств $\boldsymbol{A}$ и $\boldsymbol{B}$;

$\boldsymbol{A} - \boldsymbol{B}$ — разность множеств $\boldsymbol{A}$ и $\boldsymbol{B}$, т. е. множество тех элементов $\boldsymbol{A}$, которые не входят в $\boldsymbol{B}$;

$\varnothing$ — пустое множество;

$\boldsymbol{A}'$ — дополнение множества $\boldsymbol{A}$;

$\bigcup$ — объединение системы множеств;

$\bigcap$ — пересечение системы множеств;

$\langle a, b, \ldots, c\rangle$ — упорядоченная последовательность $a, b, \ldots, c$;

$\boldsymbol{A}_1 \times \ldots \times \boldsymbol{A}_n$ — декартово произведение множеств $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_n$;

$\boldsymbol{A}^n$ — декартова степень множества $\boldsymbol{A}$;

$\boldsymbol{N}$ — множество натуральных чисел;

$A \overset{\mathrm{df}}{=} B$ — A по определению есть B;

$A \& B$ — A и B;

$A \vee B$ — A или B;

$\neg A$ — не A;

$A \Rightarrow B$ — A влечет B;

$A \Leftrightarrow B$ — A равносильно B, т. е. A тогда и только тогда, когда B;

$(\exists x)(\ldots)$ — существует такое значение x, что ....

$(\forall x)(\ldots)$ — для каждого значения x ...;

Числовые функции и операторы:

Словарные функции и операторы:

Другие знаки:

Анатолий Иванович Мальцев
Алгоритмы и рекурсивные функции

М., 1965 г., 392 стр. с илл.

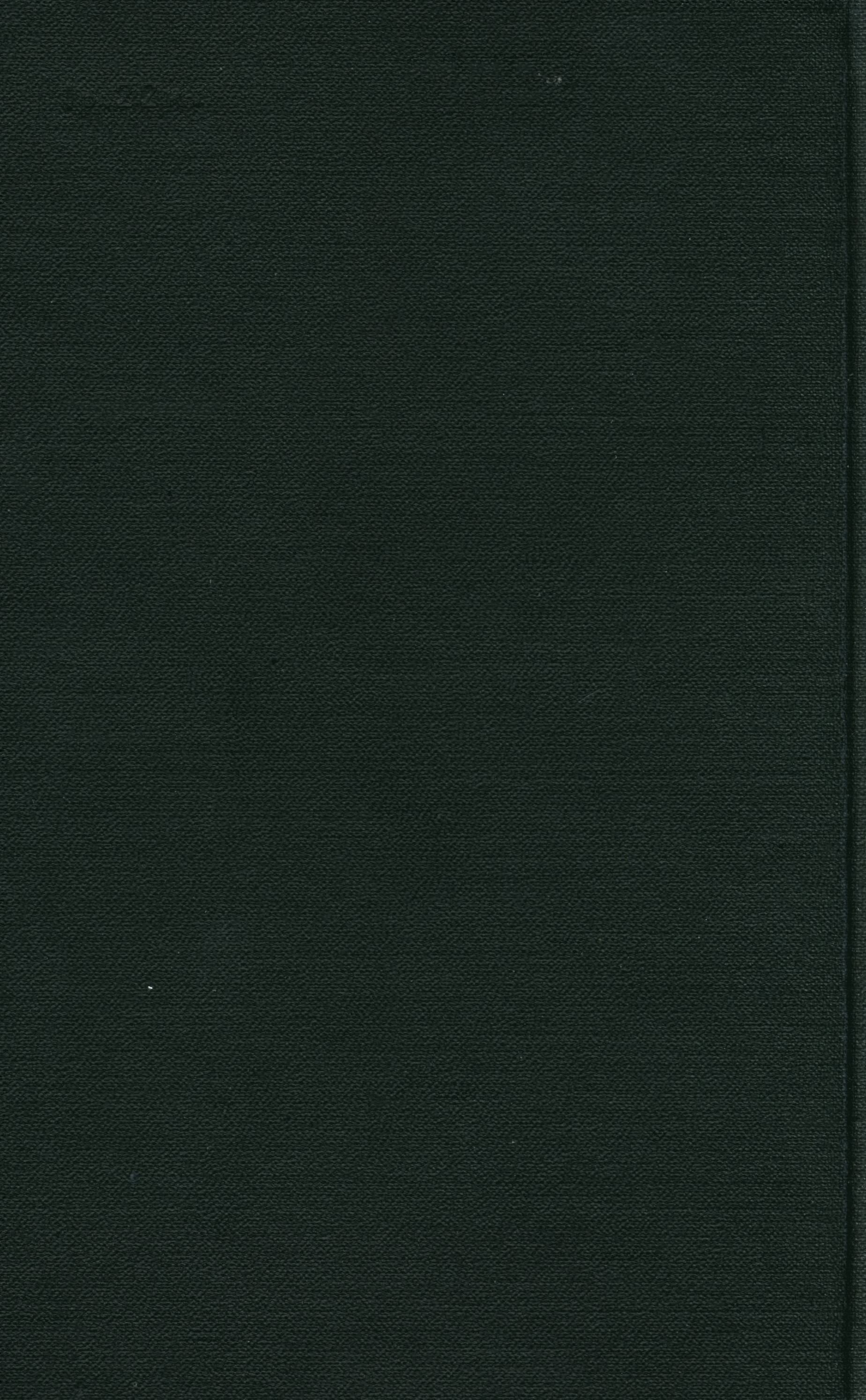
Редакторы *Д. А. Захаров*, *В. В. Донченко*

Техн. редактор *Л. Ю. Плакше*
Корректор *Г. Г. Желтова*

Сдано в набор 3/V 1965 г. Подписано к печати 21/X 1965 г. Бумага $84\times108/_{32}$. Физ. печ. л. 12,25. Условн. печ. л. 20,09. Уч.-изд. л. 19,26. Тираж 13000 экз. Т-13742. Цена книги 1 р. 22 к. Заказ № 1082.

Издательство «Наука»
Главная редакция
физико-математической литературы.
Москва, В-71, Ленинский проспект, 15.

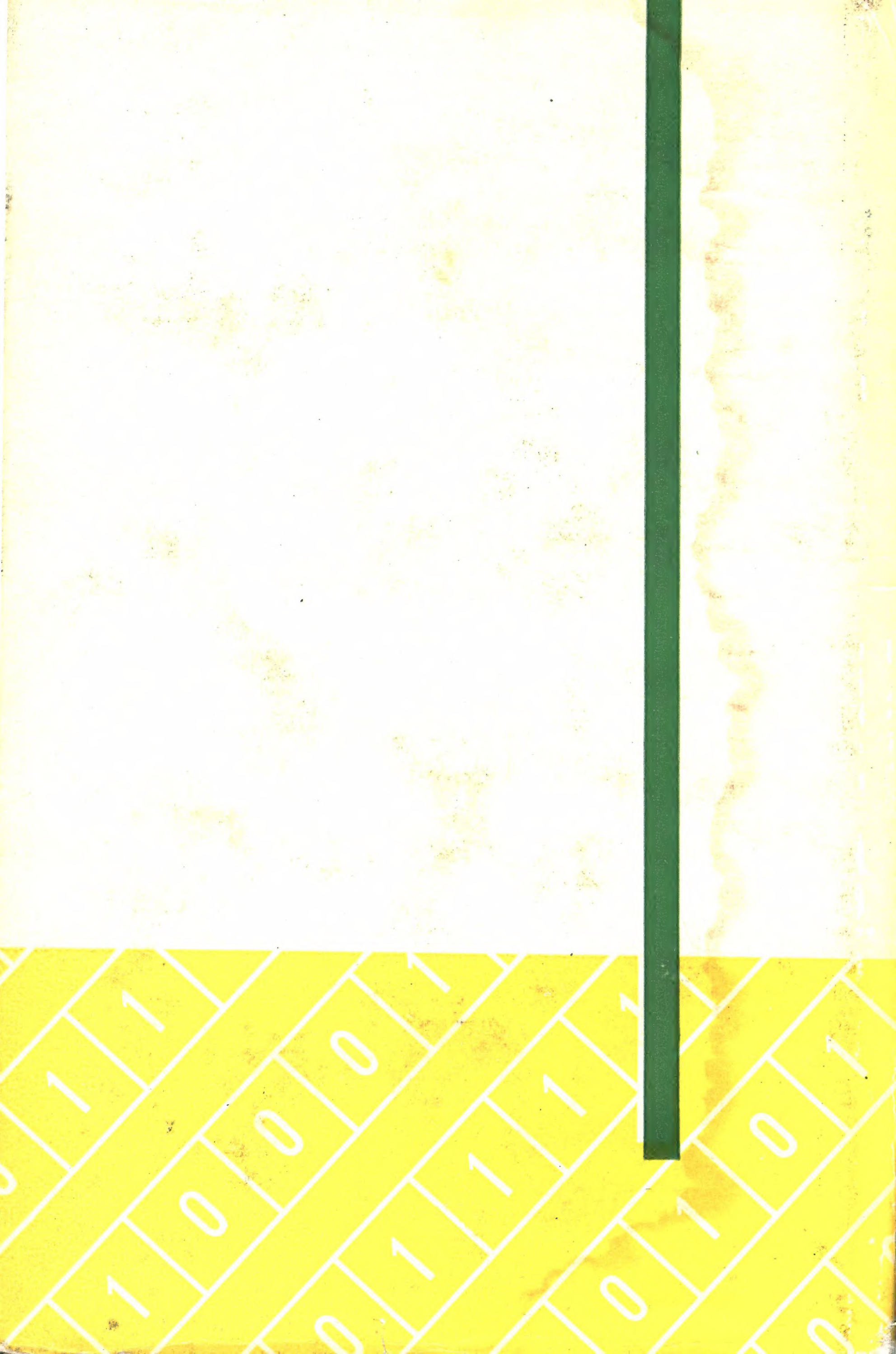
Московская типография № 16
Главполиграфпрома Государственного комитета
Совета Министров СССР по печати.
Москва, Трехпрудный пер., д. 9.



Цена 1 р. 22 к.

Алгоритмом называется предписание (программа), указывающее, какие простейшие действия и в каком порядке следует совершить над заданными величинами, чтобы получить искомую величину. Фиксируя программу и меняя заданные величины, будем получать различные значения для искомой величины. Таким образом, искомая величина будет функцией от заданных величин, вычислимой с помощью фиксированного алгоритма. Функции, обладающие указанным свойством, называются рекурсивными.

Все детерминированные вычислительные машины работают согласно алгоритмам, определяемым конструкцией машин и начальными состояниями некоторых частей машин. Поэтому функции, вычислимые на машинах, заведомо рекурсивные, что связывает теорию рекурсивных функций с теорией вычислительных машин. Не менее важное значение теория алгоритмов и рекурсивных функций имеет и для самой математики. Развернутая теория алгоритмов позволяет находить алгоритмы для решения сложных математических задач. Вместе с тем теория алгоритмов позволяет доказывать несуществование алгоритмов для ряда интересных классов задач.



А. И. МАЛЬЦЕВ

АЛГОРИТМЫ И РЕКУРСИВНЫЕ ФУНКЦИИ